'68' MICRO JOURNAL

$2.95USA

Australia	A $ 4.75	New Zealand	NZ $ 4.75
Singapore	S $ 9.45	Hong Kong	H $23.50
Malaysia	M $ 9.45	Sweden	30:-SEK

VOLUME V ISSUE II • Devoted to the 68XX User • February 1983

"Small Computers Doing Big Things"

YOUR CHOICE-smart either way

- Over 140 software driven functions
- 82 x 24 or 82 x 20 screen format — software selectable
- High resolution 7 x 12 matrix characters — P-31 green phosphor
- Upper/lower case character set — plus graphics character set
- 56-key alphanumeric keyboard — plus 12-key cursor, numeric pad
- Internal editing functions — insert, delete, scroll, roll, slide, etc.
- Parallel printer I/O port
- 50 to 38,400 baud operation — programmable
- Cursor type, cursor position, print control characters, protected fields, shift inversion, dual intensity and many other features

8212 — twelve-inch diagnonal screen or **8209** — nine-inch diagnonal screen

SWTPC

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216 (512) 344-0241

Pascal for 6809

Pascal for the 6809 is a true native code compiler. Unlike the usual P-code Pascals which run in an interpretive manner, ours produces efficient assembly language mnemonics which can be assembled and run directly. This compiler is available for both 6809 FLEX™ and UniFLEX™. Many features not found in other Pascal systems were implemented while avoiding those features completely non-standard. Features of the Pascal system include:

- Supports most of Jensen and Wirth specification
- Produces fast and efficient 6809, native code
- FLEX run-time package may be trimmed
- Double precision real numbers (16.8 digits)
- Implements scalar, subrange and structured data types
- Standard I/0 using file buffer pointers
- Dynamic storage allocation
- Ability to call other Pascal programs
- FLEX version may call assembly language programs
- Buffered or single character terminal input
- Standard math functions: SIN, COS, ARCTAN, EXP, LN, SQR, SQRT
- Random number generator function
- Many usable, sample programs included
- UniFLEX version supports:
  - Random file positioning
  - Ability to call various UniFLEX system routines
  - Ability to execute UniFLEX utility commands

Pascal on diskette for 5" and 8" 6809 FLEX is available for $200.00 The 5" version requires two disk drives. The UniFLEX version is $300.00 and includes one year of maintenance. All orders should include 3 percent for postage and handling (10 percent on foreign orders).

™FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc.

technical systems consultants, inc.
111 Providence Rd., Chapel Hill, N.C. 27514
(919) 493-1451

'68'

MICRO JOURNAL

Portions of text prepared using the following.

SWTPC 6800-6809-DMAF2-CDSI-CT82-Sprint 3
Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216

EDITOR - WORD PROCESSOR
Technical Systems Consultants, Inc.
Box 2573, W. Lafayette, IN 47906
FLEX is TM of TSC

GIMIX Super Mainframe-Assorted memory boards
GIMIX Inc.
1337 West 37th Place
Chicago, Il 60609

CONTENTS

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES
USA
1-Year $24.50 2-Years $42.50 3-Years $64.50
FOREIGN
See Page 52

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors full name, address, date and telephone number. It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands ONLY should be used (due to our proportional processor): .sp space, .pp paragraph, .fi fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continous text.

All articles submitted on diskettes should be in TSC FLEX™ format, either FLEX2 6800, or FLEX9 6809 any version.

If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). Please use a dark ribbon!

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word. 20 words or less $7.50 minimum, one time, paid in advance. No classified ads accepted by telephone.

2MHZ 6809 SYSTEMS

GIMIX offers you a variety to choose from!

38 MB WINCHESTER SYSTEM $17,498.99

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 512KB Static RAM
- ★ 8 RS232C Serial Ports
- ★ 2 Parallel Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ Dual 8" DSDD Floppy Disk System
- ★ Dual Winchester Subsystem with Two19 MB 5¼" Winchester Drives

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Debugger
- ★ OS-9 Text Editor
- ★ OS-9 Assembler

19 MB WINCHESTER SYSTEM $8998.09

HARDWARE FEATURES:

- ★ 128K Static Ram
- ★ 2MHz 6809 CPU
- ★ 19 MB 5¼" Winchester DMA Subsystem
- ★ 4 RS232C Serial Ports
- ★ 1 MB 5¼" Floppy Disk Drive
- ★ DMA Double Density Floppy Disk Controller

SOFTWARE FEATURES:

- ★ OS-9 LEVEL TWO Multi-User Operating System
- ★ OS-9 Text Editor
- ★ OS-9 Debugger
- ★ OS-9 Assembler

128KB MULTI-USER SYSTEM $6997.39

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ DMA Double Density Floppy Disk Controller
- ★ 128KB Static Ram
- ★ 2 RS232C Serial Ports
- ★ Dual 8" DSDD Floppy Disk System

SOFTWARE FEATURES: Your choice of either UniFLEX or OS-9 LEVEL TWO. Both are Unix-like Multi-User/Multi-Tasking Operating Systems.

56KB FLEX/OS-9 "SWITCHING" SYSTEM $4148.49

HARDWARE FEATURES:

- ★ 2MHz 6809 CPU
- ★ 56K Static Ram
- ★ 2 RS232C Serial Ports
- ★ DMA Double Density Floppy Disk Controller
- ★ 2 Built-in 5¼" 40tr DSDD Disk Drives (80 Track DSDD Drive Option . . add $400.00)

SOFTWARE FEATURES:

- ★ GMXBUG monitor — FLEX Disk Operating System
- ★ OS-9 LEVEL ONE Multi-tasking operating system for up to 56K of memory

WINCHESTER SUBSYSTEMS

Winchester packages are available for upgrading current **GIMIX** 6809 systems equipped with DMA controllers, at least one floppy disk drive, and running FLEX, OS-9 LEVEL ONE or OS-9 LEVEL TWO. The packages include one or two 19MB (unformatted) Winchester drives, DMA Hard Disk Interface, and the appropriate software drivers. The Interface can handle two 5¼" Winchester Drives, providing Automatic Data Error Detection and Correction: up to 22 bit burst error detection and 11 bit burst error correction. **UniFLEX NOW AVAILABLE**

Dual drives can be used together to provide over 30 MBytes of on line storage -- or use one for back-up of the other. (More convenient and reliable than tape backup systems.

#90 includes one 19MB Drive, Interface, and Software **$4288.90**
#91 includes two 19MB Drives, Interface, and Software **$6688.91**

Contact GIMIX for systems customized to your needs or for more information.

50 HZ Export Versions Available

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

SOME THOUGHTS

The name of this text file on my disk is NOTES40. To bring up some ancient history, I decided some time ago to start a FLEX User's Group newsletter. In the first year as an "Independent", I managed to get out 8 newsletters, and had some 60 or so, "members" or "subscribers". At that point, the writing was fun, but the pains of getting copies made and mailed was more than I had time for, so I contacted Don Williams to see if he would be interested in turning my newsletter into a column.

I remember thinking that I shouldn't promise to get a column out every month. Don looked at some of my early efforts, and offered to send my "subscribers" the first four issues of '68' Micro Journal in which my column would appear, as a fulfillment of my obligation, since I had indicated that the subscribers would receive 12 issues of a newsletter.

The title NOTES40 means just that. This is newsletter number 40. The first four were done with Miniflex, and I can steer anyone interested to a source for them, since a "Society for the Preservation and Encouragement of Miniflex Use in America" (I'm just kidding with the name, of course) has been established. I have the newsletters, but the software listings associated with them would be of little use to most of the readers of '68'. What I am getting at, is the question of whether more than three of you readers would be interested in a collection of about three years worth of these columns, of course complete with program listings as they have appeared from time to time.

If you might be interested, please send a quick note or card to '68' Micro Journal and express your interest. If Don Williams receives enough response, it is very likely that he will put all my efforts together and print them.

WHY I LIKE PASCAL

In 1000 words or less.... Just kidding. My arm is a bit out of joint from patting myself on the back, but I will describe what I did and why briefly should anyone care. I've been working on a machine control program. It is about 12K of object code in OmegaSoft Pascal. Compile - Assemble - Load time total about 12 minutes. I had written the program to be easily modified for a range of machines in different configurations. I put a half page of constants at the beginning. (The listing is 15 pages). Now, for each machine, I only had to edit that first half page and recompile the program, burn a set of EPROMS, and the job is done.

Sounds reasonable, and it was not bad at all. Suddenly had a mild brainstorm. I have one EPROM on the processor board addressed at F800 to FFFF, so I can have restart and IRQ vectors defined. That just takes a few bytes at the very end of memory, leaving almost 2K empty. Why not declare those constants as INTEGER variables at absolute addresses starting at $F800? Now the Assembler file I used to call START, can contain the machine parameters, entered by using FDB decimal numbers. The main Pascal program need not be edited and recompiled. I simply edit a two page Assembler program, assemble it (takes about 30 seconds) and burn one EPROM, and the machine is "personalized". If we goof, and the characteristics of the machine require a slight change in one or more of the constants, it becomes a 5 minute change rather than a half hour. (It takes a good ten minutes to burn 6 2716 EPROMS if you are all organized.)

The whole thing took about two hours to implement. A simple change of the Constant declarations to Variables, followed by an edit of the START program was all it took. The thing worked on the first try. A side benefit is that I don't have to keep a very large disk file for each machine, just a 5 sector configuration file. There is an added benefit. If there are some machine constants that might improve the performance if "tweaked" a bit, I can modify them and run an experiment "instantly" compared to the previous method.

CURED AT LAST

I have had a long standing and rather irritating problem with my SWTP system. I've had a pair of 8" drives for just two years now. When I first installed them, I noticed that I had no trouble formatting a disk if I did it immediately after turning the computer on. After half an hour, I noticed a large increase in head "re-seeks" when trying to read a disk. After a bit of experimenting, I found that I had to cool something on the DMAF board. I had been running my SWTP box with the cover off, since I change boards and connections rather frequently. I didn't have a fan handy, so I made a "stovepipe" out of a piece of cardboard and found that setting it on top of the cards, roughly over the DMAF, made the system settle down.

Not wanting to have the only computer in town with a chimney, I installed a fan in the case and glued paper inside to close up most of the holes in the ventilated top. I put the fan at the right side of the front of the box, just in front of the power supply, and left an opening at the left side so that air could be brought in there and drawn across the cards. The problem went away... for a while. It came back a year later, and got so bad, even with the fan, that I couldn't format a disk. I carefully marked the position of the three "trimpots" in the data separator circuit, and broke the seals. I found that rotating one of them slightly made the system read fine again. I think now, that that particular problem was unrelated to the long term one.

At any rate, the problem recently returned. I found, with some circuit cooling spray that cooling the 1791 disk controller made the system work again. My temporary cure was to mount the fan in reverse, outside the case and let it blow directly on the 1791. I plugged the DMAF card into the board slot where it got the most air from the fan. Since then I've replaced the 1791 and had no problem again operating with the cover removed. I agree, it doesn't look as nice, but it sure facilitates transfer of a couple of 5" drives back and forth between it and the Color Computer.

I have heard from one other FLEX user with similar problems. Perhaps some of the 1791's are just a bit marginal in some specification that is critical to the operation in the DMAF, and a few extra degrees temperature rise causes them to stop working properly. I'd be interested to hear if anyone else has had similar problems. I've been operating the system all evening with the top off, and had no problems at all.

LONG CORD

My computer is located in the basement, next to the furnace. I get a bit tired of listening to the blower kick on and off every half hour when the weather starts to get cold again. Further, I long for a change of scenery once in a while. Sometimes I am working on a program that requires 30 seconds of operator input followed by 15 minutes of compile or assemble time, a five minute test, quick edit, and a recompile. For all those reasons, I decided that it would be nice to put my terminal on an extension cord. The idea only works if I am doing a lot of editing, writing, or testing that involves only my system disk and one other. That is, it is not terribly handy to be running down the basement steps every five minutes to change disks. Frequently, however, that condition is fulfilled, and I find that I can bring the terminal up to the family room and watch TV with the family between interactions with the terminal.

I have about 25 feet of plain ordinary multi conductor wire cord between my ADM-3A and the computer. I am running the terminal at 19,200 baud, and there has not been a "burp" all evening. I've edited this text to this point and typed in several pages of a manual I am working on. Recently, I spent half a day with the terminal on the redwood picnic table on the screened porch, enjoying the great late summer weather while working on a work project.

AN APPLICATION

At least one reader has asked me to present some sort of application of a computer to the real world. A long time ago, I became interested by necessity, in the use of thermistors to measure temperature. A thermistor is made of a semiconductor material, and its primary characteristic is a resistance that changes with temperature, decreasing as the temperature increases. What is unique about the thermistor is not that its resistance changes with temperature, since all metals change in their resistance to electric current with temperature. Metals show a very small increase in resistance with a change in temperature. What is unique about a thermistor is the magnitude of its resistance change, and the fact that the resistance decreases as temperature increases. We would say that the thermistor has a large negative temperature coefficient of resistance. The coefficient is in the range of 2% to 4% per Celsius Degree.

The resistance of the thermistor may be measured by one of several means. For computer applications, it is desirable to measure a voltage with an A/D converter, rather than a current. If the thermistor is connected in series with a fixed resistor, across a voltage source, and the voltage across the fixed resistor measured, it will be found to change with temperature, since the thermistor and the resistor form what is called a voltage divider, and since one of the resistors changes value, the ratio of the voltage divider changes.

I once reasoned that there might be a "best value" for the series resistor that would yield a voltage output that is nearly linear over a specified temperature range. I wrote some equations to solve for the fixed resistor that would result in the output voltage from the divider being linear for three temperatures, the lowest, highest, and midpoint of the temperature range desired. It turns out that the equations are "solvable" easily, and that the resulting value does give a pretty good approximation to a linear output voltage with temperature. The maximum error depends on the temperature coefficient of the thermistor and the temperature range.

For a common thermistor produced by a manufacturer called FENWAL, (in particular, a bead thermistor type GA45P2), a temperature range of 15 C. to 35 C. (50 to 86 F), the maximum error is approximately 1/16 C. degree. The error curve is approximately sinusoidal, the departure from linear being positive for one half of the scale and negative for the other.

I recently became interested in the problem again, and wrote a BASIC program to calculate the best series resistor, expected voltage output, and maximum errors for a given thermistor. Thermistors have two parameters that must be known in order to perform the calculation. First is their nominal resistance at 25 degrees C. Second is their coefficient, variously expressed by different manufacturers. Fenwal uses a coefficient Beta such that the resistance at any temperature may be expressed (with R0 representing the resistance at 25 degrees C.) by the following equation:

R = R0 * EXP (Beta * (1/(T + 273) - 1/298))

The value T+273 is the temperature in degrees Kelvin (Absolute), and the 298 corresponds to 25 degrees C. in Kelvin. I'll include the program here. The voltages calculated are volts per volt applied to the thermistor resistor combination. A JPC A/D card may well be used to measure temperature with such a setup. First, the thermistor divider network output is run through an "operational amplifier" that subtracts the voltage at the minimum temperature so that the output is zero for that temperature. The amplifier has voltage gain so that the output at the high end of the temperature range is 5 volts. The temperature then, calculated from the integer value from the A/D converter for our example of 15 to 35 degrees C. (a range of 20 degrees) connected to the output of the amplifier, would be. A/D output * 20/255 + 15. To express the temperature in degrees F. you would use A/D out * 36/255 + 50. Note that the maximum error in voltage from the thermistor is less than the resolution of the 8 bit A/D for the example given here.

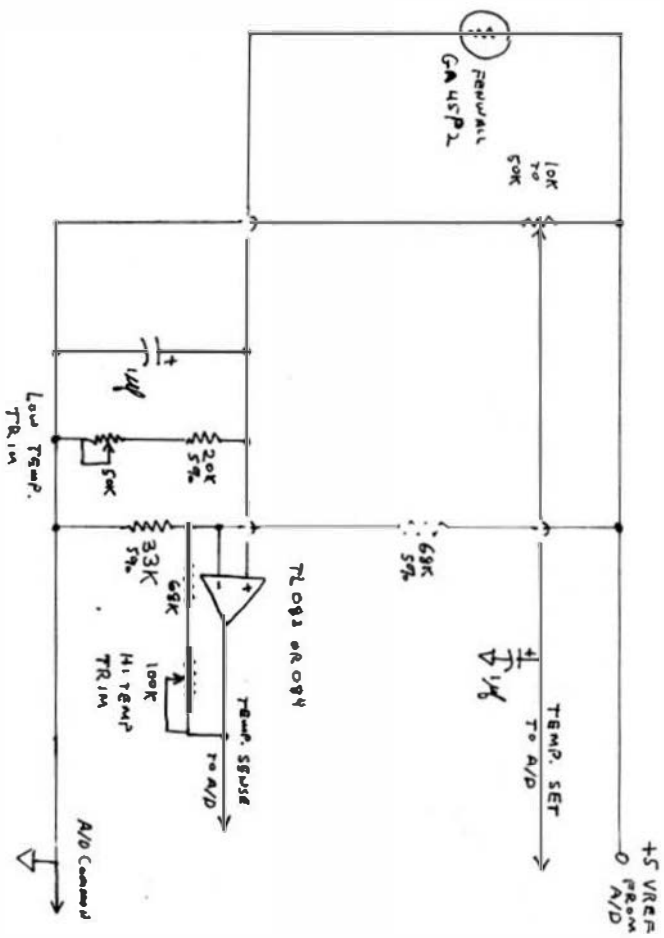
Thermistors have large tolerances on their nominal resistances, and they might therefore be in error by several degrees without calibration. The resistor in series with the thermistor should be variable, and adjusted for correct reading at minimum temperature. A lab thermometer and a bucket of water will provide a good enough reference. Ice may be used to adjust the temperature to 50 degrees F, and the series resistor set for the correct output of zero volts from the amplifier. The bucket of water should then be brought to 86 degrees F, and the amplifier feedback resistor (gain) set to produce 5 volts out of the amplifier. The unit should now be calibrated to the accuracy of your thermometer with a maximum additional error of .062 degrees.

I don't need to say that the JPC card has 16 inputs, and that you may connect several thermistors and amplifiers to several inputs to monitor temperatures at various locations. The resistance (50K) of the thermistor at 25 degrees, is high enough so that long wires won't appreciably change the calibration. Suppose you put several of these around the house, and used a potentiometer connected from reference voltage to ground on another A/D input, as a temperature "command". You could have up to 8 "thermostats" one in each room of your house and some to spare. Now, connect your computer to control motors that open and shut baffles in your hot air ducts to the various rooms in response to the difference between the set and measured temperature, and you will have a zone heating system that will save you lots of fuel all winter.

Should you not want to tie up your computer all winter, the software for this system could easily be put in EPROM and run in a "single board" computer. Output signals to open and close baffles would depend on the type of actuator you can find. Generally, just writing an "open" or "close" signal to a parallel port, could be sufficient with a little hardware to interpret the logic level at the port. I will continue next month with further descriptions of software that could be used to do this project. Of course, you would have to bypass your normal thermostat connection so that if any zone were requesting heat, you would start the furnace and supply heat to that area. To keep the furnace from cycling too often you would probably want to build in a timer such that an area would have to demand heat for a few minutes before starting the furnace. The BASIC program that I wrote for linearizing the thermistor output calculates a series resistor of the value shown in the circuit diagram, for the GA45P2 probe thermistor from FENWALL. I will include the program listing here, so you can use other available thermistors if you have the data available.

```
10  REM       THIS PROGRAM WILL CALCULATE THE VALUE OF SERIES
20  REM       RESISTOR NECESSARY TO LINEARIZE THE OUTPUT OF A
30  REM       THERMISTOR OVER A SPECIFIED TEMPERATURE RANGE.
40  REM       THE THERMISTOR PARAMETERS WILL BE ASKED FOR.
50  REM       THE MIN AND MAX TEMPERATURES CENTIGRADE WILL BE INPUT.
60  REM       THE PROGRAM WILL CALCULATE THE BEST VALUE FOR THE SERIES
70  REM       RESISTOR, AND THEN CALCULATE THE VOLTAGE OUTPUT OF THE
80  REM       NETWORK FOR MIN AND MAX TEMPERATURE, AND THE VOLTS PER
90  REM       DEGREE, ALL AS A FRACTION OF THE SUPPLY VOLTAGE.
100 REM
110 REM       THE PROGRAM WILL THEN SEARCH THE INTERVAL FOR THE
120 REM       MAXIMUM ERROR (BOTH HIGH AND LOW) AND REPORT THAT
130 REM       IN DEGREES.
140 REM
150     DEF FNT(X)=EXP(B*(1.0/(X+273.0)-1.0/298))*R1
160 REM
170 REM       298 IS ABSOLUTE (KELVIN) VALUE FOR 25 DEGREES C.
180 REM       X IS TEMPERATURE IN CENT.DEGREES, R0 IS RES AT
190 REM       25 DEGREES C.  B IS A CONSTANT FOR THE THERMISTOR.
200 REM
```

```
210  REM         SUBROUTINE TO CALCULATE BEST SERIES RESISTOR
220       GOTO 270
230       R=(RM*(RH+RL)-2*RH*RL)/(RH+RL-2*RM) : RETURN
240  REM
250  REM         START OF MAIN PROGRAM
260  REM
270       INPUT "R0 (RESISTANCE AT 25 DEG. C)",RI
280       INPUT "BETA",B
290       INPUT "TEMP. MIN.",TL
300       INPUT "TEMP. MAX.",TH
310  REM
320       RL = FNT(TL)
330       PRINT "RES AT LO LIMIT";RL
340       RH = FNT(TH)
350       PRINT "RES AT HI LIMIT";RH
360       RM = FNT((TL+TH)/2)
370       PRINT "RES AT MID SCALE";RM
380  REM
390       GOSUB 230 : REM R CONTAINS THE BEST SERIES R VALUE
400       PRINT "SERIES RESISTOR ";R
410  REM
420       VL = R/(R+RL)
430       VH = R/(R+RH)
440       PRINT "VOLTAGE (PER SUPPLY VOLT) FOR TEMP ";TL;" IS ";VL
450       PRINT "VOLTAGE (PER SUPPLY VOLT) FOR TEMP ";TH;" IS ";VH
460  REM
470       VD = (VH-VL)/(TH-TL)
480       PRINT "VOLTAGE PER DEGREE PER SUPPLY VOLT IS ";VD
490  REM
500  REM         CALCULATE WORST ERRORS
510  REM
520       FOR N = 0 TO 100
530       T = TL + (TH-TL)/100 * N
540       VT = VL + VD * (T-TL) : REM THEORETICAL LINEAR VOLTAGE
550       RT=FNT(T)
560       VA = R/(R+RT)
570       ER = (VA - VT)/VD : REM ERROR IN DEGREES
580       IF ER > EP THEN EP = ER : TP = T : REM SAVE MAX
590       IF ER < EM THEN EM = ER : TM = T : REM SAVE MAX NEGATIVE
600       NEXT N
610  REM
620       PRINT "MAX POSITIVE MEASURE ERROR ";EP;" DEGREES"
630       PRINT "MAX POSITIVE ERROR AT TEMPERATURE ";TP
640       PRINT "MAX NEGATIVE MEASURE ERROR ";EM;" DEGREES"
650       PRINT "MAX NEGATIVE ERROR AT TEMPERATURE ";TM
660       END
```



COLOR User Notes

ROBERT L. NAY
4429 Plantation Lane
Norcross, GA 30071

We are constantly receiving letters and calls with questions about the use of FLEX (or requesting capabilities that are already available with FLEX) from new Users, so I will begin a discussion of Disk Systems and the FLEX Disk Operating System this month, and complete it with a discussion of some of the more useful Commands and Procedures next month. This should accomplish a couple of things: first, to help the new FLEX Users get "up and running"; and second, maybe answer some of the "Why use FLEX?" questions many of you have.

Since I am firmly convinced that some knowledge of the "HOWS" and "WHYS" make things easier to understand, I will also provide some basic information as we go along.

FLEX Overview

FLEX really is a "Sweet" Disk Operating System, even though it may look complex to a new User who has no experience with a major Disk Operating System. The function of a DOS (Disk Operating System) is to take care of all of the Disk Operations, so that the User does not have to worry about keeping track of which File (Program, Text Material, etc.) is located where on a Disk, or where to go to get a File, etc. A GOOD DOS is effectively invisible to the User, in that he is not aware of the Disk Operations that are taking place, and does not have to change his normal procedures to fit them to the Operation of the Disk System. A GOOD DOS allows the Computer Operator to USE the System in accomplishing HIS task, while not worrying about it accomplishing it's OWN duties. The USER does not have to worry about things like losing Files be ause they were left "Open", or WHERE a File will be stored, or HOW it will be stored. Agreed, many of these features depend on the Programmer who wrote the Software that is being used, but a GOOD DOS makes the incorporation of these functions easy, so that the USER does not have worry about them. Finally, a GOOD DOS provides a "natural" Command Structure; the DOS Commands are a natural extension of the way a USER thinks, so he does not have to "learn a new language" to use the Disk Operating System comfortably. (For example, compare the FLEX "COPY.CMD" to the CP/M "PIP" Command; who would ever DREAM that PIP had any thing to do with Copying Files??) FLEX has all of these features.

FLEX Commands are entered following the familiar FLEX Prompt of three plus signs (+++). Any time I refer to "being in FLEX", it will mean that you have the +++ Prompt

followed by the cursor. A FLEX Command consists of three things; the Filename (the name of the command, program, etc.), the filename Extension (CMD for Command, TXT for Text, DAT for Data, etc.), and the Drive Number (0 for the first Disk Drive, 1 for the second Disk Drive, etc.). Also, you can use EITHER a Space OR a Comma for Command separaters or delimiters. (As we shall see, most times you only need to use the Filename, the System knows the rest of the information automatically.) For example, if you want to see a CATalog of the Disk in Drive #1, you COULD enter

```
0.CAT.CMD 1    — or --    0.CAT.CMD,1
```

and the list of the Files on the Disk in Drive 1 would be displayed on the Screen. I said COULD because you would normally just enter

```
CAT
```

and FLEX would look for a File called "CAT.CMD" on Drive 0, and would know that you want that Command to operate on Drive 1 (because of the ASN.CMD, which we will be discussing next month). The Drive Number does not have to appear first in the Command Line; it can also follow the Extension. Both of the following Commands would be interpreted the same;

```
0.CAT.CMD 1     or     CAT.CMD.0 1
```

I referred to the System Drive and Work Drive a little earlier. The "SYSTEM Drive" is the Drive containing the Disk with all (or at least MOST) of the COMMANDS on it; these will ALL have a ".CMD" Extension. When FLEX expects a Command (it expects you to tell it WHAT you want it to do first), it will automatically look for the Filename you give to have a ".CMD" Extension, AND it will automatically look on the specified SYSTEM Drive for it. When we just entered "CAT" above, it assumed that we were referring to CAT.CMD on the System Drive. Also, it assumed we were wanting it do what we requested to the Disk in the WORK Drive. The WORK Drive contains the Disk we are normally "WORKing" on. The System and Work Drive CAN be one and the same, but it is much better to have at least two Disk Drives on your System when using FLEX. There are two reasons for this. First, you will rapidly fill up even a Double Sided Double Density Disk with COMMANDS. Most Utilities, such as COPY, DELETE, CAT, etc., are short - one to five Sectors - but a lot of the Programs are large. The EDITor is forty some Sectors, the ASseMBler is forty some Sectors, STYLO is around 86 Sectors, and DYNACALC is over a hundred; not to mention some of the DICTIONARIES which run over 300 Sectors. Second, most of the Disk Writing is done to the Work Disk; if something goes wrong, flies are destroyed by Writing all over them, not by Reading them, so the Disk being Written too is more susceptible to damage (It is normally NOT a problem with FLEX; I have only had a couple of problems over two years of CONSTANT use with FLEX, where I averaged a problem a week with some other Systems I have been using).

(By the way, the Color Disk System does NOT come up after Turn-On with the Disk Read and Write VERIFICATION turned ON, like FLEX does; you need to enter "VERIFY ON" each time you bring the Color Computer up with the Disk System Installed.)

The DISK SYSTEM

Just what does the term "Disk System" refer too? The Disk System consists of at least ONE Disk Drive, the Disk Controller, the Computer, and a Disk Operating System (DOS) Program. The Disk Drives hold the Disk, rotate it at a very tightly controlled speed (300 RPM for 5 1/4" Disks), positions the Read/Write Head per the Controllers instructions, and either Reads information off of the Disk or Writes information to it. The Disk Controller is the communication link between the Computer and the Disk Drive. It executes the Computers instructions. If the Computer asks the Controller to get the information from a certain Track and Sector, the Controller tells the Drive to position the Read/Write Head at that Track Number, checks off the Sectors until the correct one shows up, grabs the information as it goes by, and tells the Computer what it just read. If the Computer instructs the Controller to Write a Sector to the Disk, the Controller tells the Drive to position the Read/Write Head at the proper Track, checks the Sectors as they go by, and causes the Drive to Write the information when the correct Sector shows up. The Disk OPERATING System (FLEX or the Radio Shack Disk System), keeps track of what is going on, determines which Track and Sector it wants to operate on, and just what information is to be written to the Disk. The DOS is the "brains" of the outfit; all of the others just follow its instructions.

But, all Computers and Disk Controllers are not alike. If they were, we would have COMPLETE compatibility between ALL Computer Systems that run FLEX, for instance. Since this is not the case, some variations occur, and each Computer System must contain routines which "marry" the DOS to the Computer and Disk Controller. Since FLEX was written to be a "general" Disk Operating System, TSC said "OK, put a character from the Keyboard HERE!", and did not worry about what kind of Keyboard was to be used. "Keyboard Drivers", short Software routines, must then be written for each Computer System that is to work with FLEX. The same goes for Displays and Disk Controllers. TSC sells a Program called "FLEX.COR" that is the "heart" of the FLEX DOS; each Computer System Manufacturer or FLEX Conversion Writer then develops Keyboard, Display, and Disk Drivers which are married to FLEX to make up the Program "FLEX.SYS" which is loaded into the Computer each time you want to run the FLEX Operating System. There are two other "System" routines that are normally found on a FLEX Disk. These are "PRINT.SYS", which marries the FLEX DOS to the Printer Output, and "ERRORS.SYS", which provides the User with an English Language translation of an error, such as "Drives Not Ready", instead of some weird message like "Error No. 4".

The Floopy DISK

Now let's look at the Fl py Disk itself, for a moment. 5 1/4" Floopy Disks come in several different flavors. You can get Single Sided, Single Density certified Disks; Single Sided Double Density certified Disks; the same versions for Double Sided; and ALL of these versions as Soft Sectored, 10 Sector Hard Sectored, 16 Sector Hard Sectored, etc., etc., etc. Well, happily, we can narrow the choice down a "chunk". First, lets see what these different "certifications" mean.

When the Disk is Manufactured, a coating of magnetic material is deposited on the surface to hold the information we want to store on the it. If the Disk is being made for Single Density use, the coating does not need to be very high quality, because there is not going to be as much information stored on a certain amount of the Disks Surface. But, if it is designed for Double Density storage, there will be a lot more information stored on the same amount of surface. Therefore, the coating needs to be higher quality (and usually, thicker). Obviously, this coating costs more. The Disk Manufacturer tests the Disks for certain capabilities, and certifies them accordingly. Luckily for us Users, the Manufacturers have developed their procedures to the point that even though they will only certify the less expensive coating for Single Density, it actually works fine for Double Density use. The other major factor that makes this possible is the development of excellent Disk Controllers and Read/Write Heads in the Disk Drives. We at '68' Micro have been using GOOD QUALITY (Verbatim Datalife, Scotch 3M, etc.) Single Sided Double Density 5 1/4" Diskettes for Double Density Double Sided operation for several years with NO problems of any kind.

With the Disk Controller *technology* of today, the critical factor becomes the Read/Write Head in the Disk Drive. That little tiny piece of hardware will make or break your whole Disk System. You have a choice; you can spend $50 to $75 more for a QUALITY Disk Drive with a GOOD Read/Write Head and pay $10 to $30 less for each box of Disks, or you can buy the cheap Disk Drive and pay enough more for Disks in the first year to have *bought* a GOOD Double Sided Disk Drive in the first place. While we are on Disk Drives, let me digress a little bit. You have probably seen the Advertisements in some of the Magazines about the "Screw" Head Movement Drive versus the "Band" Head Movement Drive. Well, the 8" Drives have almost ALWAYS used the Band type of Head control, and they have been the epitome of reliability for years. The Band Stepper System allows much quicker motion, with tighter tolerances, and is less likely to have mechanical problems than the Screw Stepper Systems.

Now to the "Soft" and "Hard" Sectored Disks. Al ost the whole world has gone to the Soft Sectored Disk for all Disk Operations. A Soft Sectored Disk has ONE small hole near the inside edge of the Diskette, called the "Index Hole". The Hard Sectored Diskette has many holes near the inside drive hole. If the Diskette is a 10 Sector Disk, there will be ten holes which indicate the beginning of each Sector, and another hole with different spacing which is the "Index Hole". If the Diskette is a 16 Sector Disk, there will be 16 holes and an Index Hole, etc. The Hard Sectored Disks were popular when Disk Systems first began to be used, but are slowly disappearing from use any more. I don't know for sure, but I would guess that the reason the Soft Sectored Disks gained the upper hand was due to the development of Hardware and the flexibility they provide. (For example, FLEX uses 10 Sectors per Track for Single Density operation, and 18 Sectors per Track for Double

Density Operation. Not only that, but Track 00 is ALWAYS written in Single Density Format, while the rest of the Disk is Double Density. This would not be possible with Hard Sectored Diskettes.)

At this time, the only Disk System I know of for the Color Computer that uses the Hard Sectored Diskettes is the one that was developed by Tallgrass and is now being sold by Cer-Comp. It is a good System, but YOU CAN NOT USE IT FOR ANYTHING EXCEPT THEIR SOFTWARE; you can not run Radio Shack Disks, FLEX, or anything else. Also, NO SOFTWARE written by ANYONE ELSE will run on that System unless it is transferred over to Hard Sectored Disks first, which is not easy to do. ALL of the other Disk Controllers use Soft Sectored Disks, so the only changes that need to be made to transfer Programs between different Disk Operating Systems is to write a Software Routine that will interpret the way the Information is stored on the Disk to something the other System can understand. To transfer between Soft Sectored and Hard Sectored Disks, Different Disk Controllers would be required, along with the different Software. You can store a little more Information on a Hard Disk than you can a Soft Disk, because you do not have to use some of the Disk's Surface to Write the Sector Number on it, but the difference is small.

In summary, there are a lot of different types of 5 1/4" Diskettes, but our experience has been that all you need is Single Sided, Double Density, Soft Sectored Diskettes, which can be used with all Disk Systems including 80 Track Double Sided Disk Drives, for Single and Double Density operations.

Making a NEW DISK

Now that we have some background information, the first thing we will need to do is make up some NEW Disks to use with our Disk System. When you get a new disk out of the box, it is like a new Tape, in that there is NO information stored on that Disk. This is no problem with Tape, because each Program is written to it in SEQUENCE, and the only way you have any access to them, other than reading through each one until you get to the one you want, is by the use of the Counter on the Tape Unit. This in not controlled by the Computer, so there is no need to put any "Computer Readable" Information on that Tape (all of the "Formatting" required is accomplished by writing certain Information at the beginning of EACH Program, or File, as it is written to the Tape). With a Disk System, we want the System to manage all of the work for us, so we must write some things on the Disk Surface so that it can tell WHERE the Disk Drive's Read/Write Head is located on the Disk's Surface. Once it knows that, it can then Inform the Disk Controller where to move the Head to accomplish the specific task we have "Commanded" it to do. The DOS must be able to tell what portions of the Disk already have Files (Programs) on it, where they are, and what areas are still available for use. All of this Information must be kept on the Disk. Therefore, the DOS allocates a small portion of the Disk's Surface for recording this Information, so that it knows WHAT and WHERE the information is stored on ANY Disk that is Installed in a Disk Drive. FLEX uses the First TRACK (Track 00) on the Disk for this purpose; Radio Shack uses Track 17.

The FIRST thing we need to do with ANY Disk System is make up some new disks for use with our specific Disk System. Radio Shack can not read FLEX Disks, nor can FLEX read Radio Shack Disks, without special Programs to make the conversion for the different Systems. Not only that, but different Computer Manufacturer's FLEX Systems can NOT read some types of FLEX Disks made by other Manufacturers, such as Double Sided, Double Density; they Write DIFFERENT Information on the Disks (because of the different Disk Drivers and Disk Controllers). TSC specified that the "Standard" for ALL FLEX Systems would be a SINGLE SIDED, SINGLE DENSITY, 35 Track (or 77 Track if it is an 8" Disk) Formatted Disk. This allows a Software Manufacturer to put his Software on a Disk and sell it to ANY FLEX User, and know that he would be able to read it; that User then needs to transfer it to HIS System Disk so he can make normal use of the Program.

There is two reasons for making this transfer. First, the original Disk should be used as a MASTER DISK, and used as little as possible to protect it (in fact, if at all possible, it is STRONGLY recommended that you Write Protect the Disk by covering the Notch on the edge of the Diskette, AND NEVER REMOVE IT; then, it is almost IMPOSSIBLE for something to happen to the Disk when it is in the Disk Drive). Second, you are normally going to be using the Double Density format for normal operations, because you can get a lot more information on a Disk, and the MASTER Disk will be Single Density.

When we Format a Disk, the Computer instructs the Disk Controller to write Information on the Disk's Surface so that the Disk System can read the Track and Sector that is passing under the Read/Write Head at any Instant. It ALSO Initializes the Free Sector Map, Directory, etc., so it can keep track of the amount and location of the Files that are on that specific Disk.

How can we FORMAT a Blank Disk so we can use it? That depends somewhat on the Individual System you are using, but the general procedure is basically the same on ANY System. The Diskette's "Index Hole" gives us a starting point to work from; the Disk System now knows where a certain POINT is on some Track on the Disk. Also, the Computer can tell the Disk Controller to tell the Disk Drive to GO TO any specific TRACK on the Disk, so it starts off by going to Track 00 (the First Track on the outside of the Disk). ow the Computer knows EXACTLY where the Read/Write Head is located. The Formatting Program starts from there, and Writes the Sector Numbers and other Information on the Disks Surface as it rotates; then steps to Track 01 and does the same thing, then to Track 02, etc., until it has written Information on the whole Disk. This is strictly a timing function; we know the Disk is rotating at 300 RPM, and we know how many Sectors we want to write on each track, so we know how much time to wait between each Sector that we write to the Disk's Surface. In addition, the Computer is going to need some time to "digest" the information it gets from the Controller, or to gather up what it wants to be written to the Disk, so we need to provide some time for this to take place between Sector accesses. If the Disk System wants several Sectors worth of Information, we want to set up the timing so that about the time the Computer is finished storing a Sectors worth of Information into Memory, the Disk will have rotated to t e next Sector we want to Read. This is called "Sector Interleaving". Normally, we will Format the Disk so that we Write Sector Number 1, then skip two or three, and Write Sector Number 2, etc. These "Logical" Sectors will be interwoven among the "Actual" Sectors so that we are not waiting on the Disk to get around to the next Sector of Information that we want to Read or Write.

Once we get all of this written to the Diskette, we then go back and Read that information. This allows the Program to see if what it told the System to Write to the Disk really DID get written to it. When you NEWDISK a Disk, you will notice that the Drive Steps right along for a little bit, then comes back and tells you it is "Verifying" each Track and Sector, which takes a lot longer. It is reading the Disk and comparing that to what it knows SHOULD be written on it. NOW, all that is left is the "housekeeping"; writing the Boot Information, Free Sector Map, Directory, etc. Should FLEX find a bad Sector, it will notify you and remove that Sector from the Free Sector Map, so that it will not attempt to use it for File Storage later.

OK, now we have the Disk Formatted. Most Disk Operating Systems work with two different types of Disks; a SYSTEM Disk and a DATA Disk. Whats the difference? The SYSTEM Disk normally has the "Boot Program" and Commands on it. The Radio Shack Color Disk System contains this Information in ROM on the Disk Controller, so it does not use a System Disk, it only uses Data Disks. All of the other Disk Operating Systems that I am aware of require a System Disk. The "Boot Program" is a short Program that loads the Disk Operating System ("FLEX.SYS" with the FLEX DOS) into the Computer so that it can be used. This short Boot Program is located on the first two sectors of Track 00 on FLEX Formatted Disks. Computers designed to use the FLEX DOS have a short routine in ROM which loads these two Sectors into the computer and transfers control to that Program, which then loads in "FLEX.SYS" off of the Disk. The Color Computer Conversions work a little differently because we are not working with a Disk System that was designed to use FLEX; we use a short BASIC Program which gets a small program which has been stored on Track 17. This Program shuts off the Radio Shack ROMS and turn on the 64K RAM (so we have RAM to store FLEX in), and then loads "FLEX.SYS" off of the Disk. The difference between DATA and SYSTEM Disks is that the Data Disk does not have the Boot Routines, FLEX.SYS, or the Commands, because it will NEVER be used to get the Computer up nd running with the FLEX DOS. We will already be running under FLEX (or whatever Operating System) when we use these Disks. Making up a Data Disk simply leaves more room for Program or File Storage, because room is not taken up by System Programs or Commands.

Now that we know a little more about how the DOS works, and how the Disk is Formatted, we can understand WHY this next step in making up a New Disk is so Important when using the Color Computer FLEX Conversions. (I will be referring to the Data Comp Version 2.0 of the FLEX Conversions, but you will find similar commands with the others, and for the SAME reasons.) The VERY FIRST thing you MUST do AFTER you have formatted the disk (using the NEWDISK.CMD), is run the "MAKESYS" Program. MAKESYS.CMD has three functions:

1. It changes the formatting of Tracks 17 and 18 from a FLEX Format to the Radio Shack Disk Format.
2. It copies the information that is in those two Tracks from the Disk containing the MAKESYS.CMD to the new Disk.
3. It adjusts various portions of Track 00 so that FLEX will know that these two Tracks are already used, and will leave them alone.

READ ITEM 2 AGAIN! Note that MAKESYS.CMD copies "THE INFORMATION FROM THOSE SAME TRACKS ON THE DISK CONTAINING THE 'MAKESYS.CMD' TO THE NEW DISK". This means two things: one, the Disk containing MAKESYS.CMD must be a "Bootable" Disk (i.e., you can use it too 'RUN"FLEX' and get the FLEX System up and running on the Color Computer); and two, even though you can copy the Data Comp MAKESYS.CMD to other Disks, they MUST HAVE HAD "MAKESYS.CMD" RUN ON THEM for it to work correctly in making a new Disk. The simple solution is to ALWAYS run MAKESYS from the System Disk you just used to 'RUN"FLEX'. Its operation is also the reason you should run MAKESYS.CMD immediately after NEWDISKing a Disk; too let MAKESYS adjust the FLEX Control Sectors on Track 00 BEFORE you start putting FLEX Files on the Disk.

To summarize the procedure for making new Disks for use with the Color Computer Disk Conversions:

1. NEWDISK a blank Disk (or one that you do not need any more information off of - NEWDISK WILL DESTROY ANY INFORMATION THAT IS ON THE DISK).
2. Run the MAKESYS.CMD on the New Disk you have just Formatted from the System Disk you used to get FLEX running.

NOTE: If you only have a single Disk Drive, both of these Programs (NEWDISK and MAKESYS) allow you to change Disks to get the New Disk in the Drive. The NEWDISK.CMD loads the Program completely into Memory, so as soon as it starts asking questions about how many Tracks, Single or Double Density, etc., you can pull the Disk containing the NEWDISK.CMD and insert the New Disk. MAKESYS.CMD realizes the problem when you tell it that you want to MAKESYS the same Drive that it lives on, and it will tell you when to change Disks.

--- RLN ---

COLOR-IBM

I don't like IBM mainframes. I'm not comfortable in that atmosphere; I know I'm not in control. The University, my employer, has two IBM's linked together. It's just too much. None seems to be able to explain their strange little quirks; they seem to go out of their way to frustrate me. I saw this movie about a bad man who took over the MCP of a computer. I surmised that this must have happened at the University. I had to do something.

I have a Color Computer with 64K and disks. It is a wonderfully capable machine; I'm comfortable and happy with it. Could I use it to take on the mighty IBM's? I could work at home unobtrusively gaining access to their secrets. In order to communicate with them, I needed a modem and a smart terminal package for the Color Computer. I bought Tandy's Modem I and their Videotex rompak. The modem worked fine; the software was lousy. I next bought the disk version of Elgen Systems' COLORCOM/E. What a great package! "Hello, IBM 4341, this is a friendly Color Computer." "Password and ID number; be quick about it or I'll disconnect you" "Log ID, password, ID#." All right, logged on!

But then I got sidetracked. The statistics student in the house was getting frustrated punching and running card decks at the Computer Center; she was forced to wait in line at the printer for her output to see what mistakes had been made. She begged me to figure out a way to write and edit SAS programs from home. That sounded easy enough at the time. We would stay at home and use the Color Computer as a terminal for the IBM.

SAS is the acronym for the Statistical Analysis System, copyrighted by the SAS Institute. It is a terrific package that does everything in the book, including graphs, but is relatively easy to use—with cards. Statistical problems get big in a hurry; often 300K of memory are required for handling matrices. The number crunching requirements can also become formidable. Statistics are not a job for small microcomputers. The IBM 370/158 executes SAS in a batch mode; input is normally with cards. The phone line connects us to the IBM 4341 which operates in an interactive mode, using the Conversational Monitoring System or CMS. So we would start with the COLORCOM software, create programs in the CMS editor and then send them over to the IBM 370 for processing. SAS would have to be fooled into thinking that it was receiving cards, especially the job control cards required in batch mode.

The folk in the Computer Center assured me that it was easy. When they tried to show me how to do it, they could not get the output from the SAS program back to my virtual machine. It just disappeared! "Hey, no problem", my IBM-loving friends at the Computer Center assured me. We'll just put in a destination card to route the output back to the virtual machine; "DEST PEROTTI" will do the trick. For two days the high and low priced help in the Computer Center muxed around with my little problem! Nothing. Then, inexplicably, "DEST PEROTITI" ran like a charm on the third day. Typical.

Here is the process writ large. First fire up the Color Computer; load and run COLORCOM/E. Dial up the Computer Center and log on. Use the CMS editor to create the SAS program, saving it on the IBM disk. The first cards in the program will be the JCL (job control) cards for SAS. Submit the SAS program (file) to batch mode and SAS where it will enter the queue and eventually run. The DEST card will redirect the output back to the virtual machine's reader. Get the output file out of the reader and onto the IBM disk for safekeeping. Move the output file to the editor and check out the results. If there are errors, fix the program file and resubmit it to SAS. If the output passed muster, print a copy for yourself at the Computer Center. Then, with the output in the Color Computer's memory, logoff and print it at home to review it. It is pointless to save the output listings to the 80-C's disk, since the output is large and a copy exists on the IBM disk.

Sounds easy, doesn't it? The Color Computer and COLORCOM/E performed flawlessly. The problems exist with the IBM environment; I would guess that the linkage does not work right. The major frustration with this process is that the Computer Center experts do not understand their setup. As a result, when problems arise, they do not know what to do. So together we experimented; we still are experimenting. Maybe this little article can save you a little grief with your local IBM.

COLORCOM/E is said to be a smart terminal package. After I purchased the software, I figured out what "smart" meant. Computer terminals are said to be "smart" or "dumb"; those adjectives differentiate between terminals which have or lack display features, such as screen editing or highlighting. Smart terminal packages for microcomputers indicate those which facilitate interaction with the host computer. Good software permits the user to download to disk or to print what is received from the host. It permits files or messages to be sent to the host. It is able to be configured for the protocols of different hosts. If you dial up Compuserve on a Sunday morning to see what is in the "New York Times", you could read it with a dumb software package. With smart software you could load the paper into your disk, logoff, and print it out at your leisure.

The COLORCOM/E, though relatively inexpensive at $59, is smart. It includes an initialization subroutine which sets the parameters for the modem, the printer, and the host computer. Thankfully, a standard exists for 300 baud telephone communication; it is the Bell 103 protocol and even the IBM's use it. The default values for the modem worked fine, except that half-duplex was needed rather than full-duplex because the IBM does not echo back what it is sent. Tandy's Videotex software only runs in full-duplex; as a result you cannot see what you have typed. I had my Epson MX-80 running at 2400 baud, but the maximum for COLORCOM/E is only 1200 baud, so I had to reset the switches. The normal baud rate for 80-C printers is 600 baud; the default is set to 300 baud and must be changed. Also watch out for how many data bits your Color Computer sends to the printer; the early 80-C's sent 7 bits, the newer ones with the Basic ROM 1.1 send 8 bits.

The initialization subroutine, "Setinit/xxx", has other nice features. My favorite is the ability to define two messages which can easily be sent to the host computer. Since I cannot type under the pressure of the IBM's gaze, I often mess up the log on procedure. With COLORCOM/E I define the messages as my ID, password, and account number. After being connected to the IBM, pressing downarrow (the control key) #7 transmits the first message, downarrow #8 the second. It's fast and error free; nice feature.

Even nicer is the ability to transmit files to the IBM line by line; the file is loaded into the transmit buffer, each line is sent by pressing downarrow #1. Instead of carefully typing in commands for the host computer, a previously prepared file of commands can be sent one by one. This feature is especially helpful for us poor unannointed amateurs, trying to work our way through the IBM labyrinth, where one mistyped character spells doom. Files can be loaded into the transmit buffer even when on-line. With Compuserve, for example , you can quickly get to the location you are seeking. The commands would have been loaded into a file beforehand; when you run COLORCOM/E, you immediate the file into the transmit buffer. After you are connected with the Compuserve Computer you transmit the commands one by one by hitting control #1 over and over. With the IBM you can create and send a collection of commands to set up the terminal and disks the way you like; you can "CP spool" this, "access disk *" that, or even input the dreaded Job Control Cards into a program you are creating. To be fair to the IBM, it is easier to create and execute "Exec" files or macros on your disk there. COLORCOM/E also permits a file to be transmitted as a whole package. So, for example, after creating a new file in the IBM Editor, I set the editor to input status and then pressed the control #2 key, transmitting an entire file to the IBM file. Sometimes that procedure worked, and sometimes it literally blew away the Editor and the CMS operating system. I decided that IBM's prefer to be fed in small bites; my file was too much too fast and it made the poor thing gag.

COLORCOM/E uses two buffers of adjustable sizes. The transmit buffer, mentioned above, can either be loaded from a disk file or it can also be written to in the "Advanced Entry Mode". Those wonderful words refer to the mode running in the tim period right after power-up, before the other computer is phoned. This mode is also the best way to send commands to your printer, again using control #1. The other buffer, not surprisingly, is the receive buffer which stores everything sent or received during the session. Even with 32K, the receive buffer fills up quickly. Control #4 displays the amount of room left in the receive buffer. Control #3 "freezes" the buffer, maintaining the current contents and "freezing out" newly received stuff from the host. The buffer can easily be dumped to disk, cleared and refilled, etc. After logging off the contents of the receive buffer at your disposal. The up and down arrows can be used to scroll through the buffer; portions of it can be marked for printing or saving to the disk.

Computer Centers are not nice places. They contain a strange mixture of self-assured priest-programmers and crazed students, mostly the latter. Lots of frustration and anger are exhibited at the Computer Center. Angry graduate students, much fiercer than undergraduates, crowd around the printers, pushing and snarling at one another. They wait. They know how stupid they are; they are fully aware that their program is probably wrong and will not run. They wait for printouts which are snatched from the operator's hands, then passed from one disbelieving student to another. The printout's creator is the one who shrieks and curses. Kiss your Color Computer for saving you from this. Print at home!

Avoiding the Computer Center is what it is all about. Now maybe the cathartic behavior described above is healthy; maybe the Computer Center provides an isolated outlet for hostility and frustration, enabling the graduate students to adapt to a normal existence which is appropriately subdued and obsequious. The fact is that interminglilng with them in the Computer Center in their debased condition is frightening and should be avoided. Since all of us are fallible and inclined to computer errors, it is especially sweet to discover one's own folly on the Color Computer screen in the privacy and security of the home. Eventually even IBM programs can be persuaded to run, often creating voluminous output listings. Well, you can have it printed out down by the bestial students or, if you have lots of time and patience, you can print the final results on the little printer that sits next to the Color Computer.

Printers are tricky little devices which have their own microprocessor which responds to funny control characters. The Color Computer doesn't even have a control key, how can the printer be communicated with? The COLORCOM/E package uses the downarrow key for a control key. When the control key is used in conjunction with the 53 keys on the keyboard plus the shift key, 127 different characters or codes are then available to be sent to the printer. Since most IBM programs print out their results with 132 characters per line, the home printer will need to be set for 132 character printing. The Epson Mx-80 is set by being sent a control O (letter "O"), which is equivalent to Chr$(15)in Basic. Since escape is defined by COLORCOM/E as the control+clear key sequence, the condensed print is cancelled with control+clear followed by "F" (enter). Emphasized print, for those of us too poor to afford a new ribbon, is control+clear "E" (enter); double strike is control+clear "G". The Epson printer will print the graphs from the IBM if it is set to 132 characters per line and if you use the IBM Editor to shift the file one space to the left; SAS and SPSS really print 133 character lines.

If you want to get tricky and edit one of the files that you received from the host computer, that too is possible. The receive buffer is saved as an ASCII file by COLORCOM, so both Nelson's Superwriter and Cognitec's Telewriter can be used to edit and format these reports from the big IBM. This is a good way to fix up the results from statistical programs that were not as good as you had hoped. But, since none trusts results which are not printed on the wide green and white sheets, you will have to transmit the edited file back to the IBM for reprinting. Little joke, there.

The above is a terribly simplified account of what can be accomplished by a Color Computer function as a terminal for an IBM mainframe. The IBM mainframes remain under the control of some demonic force; but I have not yet been able to regain control of the MCP for the good guys. I still am occupied by SAS listings which

cannot be true. It is so nice and quiet in my study; I feel protected from the hostile IBM environment. Even bizarre mistakes don't make me angry, I have hope that I will eventually be able to fix them. But I too am a dumb student, unable to comprehend the programs I write and submit. I strongly recommend discovering your mistakes on the Color Computer screen minutes, rather than hours, after submission.

EJ Perotti
163-D Pine Grove Hts.
Athens, Ohio 45701

COLOR-EXP

COLOR COMPUTER EXPANSION

When you tire of games and other piddling activities with your color computer, there is a wealth of very useful and serious programs out there that can run on the color computer. For instance, those who have upgraded their color computer to run FLEX™, with either the DATA-COMP or FLIF version (don't know about the Spectral Associates version, haven't seen it yet), have the wherewithall to run almost all of the serious software that the larger 6809 computers run. But there are some serious drawbacks to running heavy software on the color computer. One of the worst is the lack of expansion capability for the color computer. The **General Automation Expansion Interface** helps to solve this important shortcoming of the color computer.

The system sent us for review consist of the two shelf CX-3001A Aluminum Chassis and the CX-2001A Expander Card. These items along with your 32K or 64K color computer, Tandy disk controller and one or two (or three or four) disk drives, transforms the system into a very powerful and versatile computer. You will have a parallel printer port (save the $125.00 plus for the Epson serial interface) 64K memory access and a buffered expansion bus. All this without hampering the original Radio Shack disk system, cassette port, joysticks or any other regular color computer function. Yep, your color computer can talk to a modem and printer at the same time. Now that is **EXPANSION** with a large **E!**

64K Access Circuit

The Expander Card allows a 32K (Rev-E) color computer to double the useable RAM. However, I should warn you that many of the Radio Shack 32K upgrades use partial defective chips (64Ks that are mapped for the good 32K portion) and you should always run the memory test outlined in a previous issue of 68 Micro Journal, to make sure that all the memory is functional, see April 1982 issue 68 Micro Journal for more on this and the memory test.

For those who have used the Data-Comp CPI 64K upgrade ($99.95 with complete instructions for all series of color computers) the risk of possible memory failure is greatly reduced, as Data-Comp CPI ships only new, prime, commercial grade, high speed 64K RAM chips. The failure percentage of those who have expanded the Radio Shack 32K to 64K has been rather high, due to their use of graded RAM chips. So if your full 64K is good memory, you now have a **real** computer and at a very attractive price.

For the BASIC programmer this means an additional 8K of RAM for PEEK, POKE and machine language routines. Machine language programs can use all of the 64K memory. Also you will be ready and **able** to run all of the newer languages and software appearing for the 6809 (see any issue of 68 Micro Journal).

The Expander Card parallel port accepts any standard centronic-type device (printer, etc). This frees up the color computer serial port for a modem or other serial device. Also for those who have any of the Radio Shack printers that run with the Model III the Expander Card accepts the printer and cable without any hitches.

For those who desire to interface other parallel devices to the Expander Card the operation is straight forward. It has a 6821 PIA with 8 input and 8 output data lines, as well 2 output control lines and 1 input control line.

As an expansion bus

It is simple to add other devices to your color computer as the Expansion Bus has buffered microprocessor signals for up to 7 additional peripheral cards. General Automation promises an impressive array of additional devices for this system. Among them are speech systems, A/D converter, D/A converter, TV digitizer and more. If the prices are in line with the price of the basic system, it should make low cost computing available to all color computer owners.

Furnished software

When you buy the Expander Card you get (source) BASIC compatible line printer software (parallel) FREE. Also provided is a utility program that allows RS BASIC to use the expanded RAM. In addition to the above they provide 3 short demo programs for the system.

Documentation

The documentation is very complete and even the novice color computer owner should experience NO difficulty in plugging things together and be up and running in a few minutes.

In addition to the software indicated above, the following documentation is included.

1. CX-2001A Installation Guide
   This consist of unpacking instructions and complete, step by step, instructions for assembling the Expander Card and disk controller card to the system. They are very complete and include large, well drawn pictorial aids in putting everything in its proper place.

2. CX3001A Expander Card Tech Manual
   The technical manual is also complete and covers the following features of the system.

   a. Memory mapped I/O locations: This list the dedicated memory locations.

   b. 64K Memory Access Circuit: Explains the 64K feature as relates to the 32K Radio Shack revision-E. This allows the full 65,024 RAM locations to be used.

   c. parallel Port Technical Description: A run down of the parallel port (printer) input and output capabilities. Includes memory locations occupied by the parallel port.

   d. An additional section is given for those who desire information on the expansion bus. This covers a technical (but understandable) description of the bus and the header and cable connections.

Power

The power for the expander card is supplied by the color computers +12 volt supply. Claimed power consumption for the expansion interface and disk controller is well below the Radio Shack maximum stated in Radio Shack documentation. Additional GA expansion and device cards will contain their own power supply and will not further load the computer power supply.

Conclusions

When we received our system for review we unpacked it and set it up according to the furnished instructions. The total time was about 20 minutes. We plugged everything up and hit the master on/off switch (we use a strip-line cord, everything is plugged into one cord and turns off/on with one switch). The system came on line and everything worked as promised.

The only thing that we do not like about the system is that some hash is picked-up by the added cable lines and gets into the TV. We added the DATA-COMP CPI **"SCREEN CLEAN $39.95"** and the hash went away. If the small amount of hash is a bother I would recommend the "Screen Clean". It plugs into the back of the color computer and also isolates the color computer from the TV (could save many bucks if something ever shorted, in the wrong place). It requires less than five minutes to install and does not require dissambely of the color computer.

All in all it is an exceptionally economical way to expand the color computer for more serious computing. The sturdy aluminum chassis is both functional and easy on the eye. The software included leaves little for the average user to do, except have fun. The documentation is complete and better than some we have seen. For quality and price, it is a excellent addition for your color computer. It will allow you to expand and keep abreast as additional expansion devices become available. No doubt we will see others market devices for this system. The special "introductory" price for the complete system (CX2001A Expander Card, CX3001A Chassis and the CX-2401A Extension Cable) is quoted at $199.95 as a package. When ordering tell them you saw the "special" in **68 MICRO JOURNAL.**

For additional information contact:

GENERAL AUTOMATION
9600 Roosevelt Blvd., Suite 100-LL
Philadelphia, PA 19115
(215) 934-3758

A 68 Micro Journal Lab Review ---

Editor's Note: For those who experience the screen hash as mentioned above, the following may help to eliminate the problem, according to an update to the manual received after the review was finished.

"Most of the noise can be cleaned up by proper placement of antenna cable." Also they recommend that the cable between the computer and the TV (antenna cable) be rolled up into a 4" diameter loop midway between the CC and the TV. Then place the loop against the flat side of the chassis. Visible results should be seen. In addition, the loop and the ribbon cable may covered with metal foil to further reduce the interference.

It is our experience that this occurs to many devices attached to the Color Computer. In our office we have noticed that the interference is somewhat different for different combinations of Color Computers and TVs or video displays. We have used B/W and Color TVs, Color Video Monitors and RGB Color Monitors, the difference is noticeable with each. For those who cannot eliminate the interference by the above suggested methods, the 'Screen Clean' is recommended. It should be noted that this problem is primarily a result of the Color C puter and the Monitor, and not the attached unit. And we know many who do not feel that the interference is worth messing with, it is to a degree, a matter of nit-picking.

DMW - - -

INDEX 1982

INDEX 68 MICRO JOURNAL 1982

The following is an index of articles and other material published during the year 1982. Some back issues are still available for $3.50 each plus mail and handling

SWTPC Annual Meeting, Staff
COLOR Clinic, DI Stefano
OS9 Seminar, Staff
COLOR Computer - Amateur Radio, Abrams
Home Accounting Program, Watson/Brady
Debug Motorola D2, Maul
Bit Bucket - Help - Classified

August

FLEX User Notes, Anderson
COLOR User Notes, Nay
'C' User Notes, Commo
FMATE (review), Abrams
COLOR Clinic, DI Stefano
DUB (review), Lewis
Stock Report (review), Tucker
Home Accounting Program II, Watson/Brady
ARCADE-50 (review), Snyder
Diskfix9, Hartman
Bit Bucket - Help - Classified

September

FLEX U er Notes, Anderson
COLOR User Notes, Nay
'C' User Notes, Commo
COLOR Clinic, DI Stefano
TELECON 'C' (review), Urie
Simple Winchester Interface, Zeff
Speak and Sing, Review
6800 to 6809, Pass
Home Accounting Program III, Watson/Brady
ET3400 BASIC, Wolach
Bit Bucket - Help - Classified

October

FLEX User Notes, Anderson
COLOR User Notes, Nay
'C' User Notes, Commo
68 MICRO JOURNAL Bulletin System, Staff
Winchester Backup, Staff
COLOR Clinic, DI Stefano
Home Accounting Program (final), Watson/Brady
B/U Recover, Pass
Debug Motorola D2, Maul
Low Cost Winchester, Graves
DMS Notes, Adams
Echo to the COLORAMA, Melbarde
Bit Bucket - Help - Classified

November

FLEX User Notes, Anderson
COLOR User Notes, Nay
'C' User Notes, Commo
Reviews, Staff
Dual Serial Card, Review
Music Board, Review
CC FORTH, Perotti
Low Cost Winchester (final), Graves
FLEX Linked List, Mansfield
FD88 Development System, Review
Typos, Zoeller
Bit Bucket - Help - Classified

December

FLEX User Notes, Anderson
COLOR User Notes, Nay
Bombed, Staff
Cheap Talker (CC), Kelty
HI-Res COLOR Graphics, Hunt
SOSBOC - FLEX (CC), Lyon
SPELLB, Review
Virtual Memory +, Scudiere
BASIC to PASCAL, Anderson
Help-Me, DMW
DYNAMITE PLUS, Fisher
Bit Bucket - Help - Classified

SWTPC - Southwest Technical Products
CC - Color Computer

Note: Many of the items published under the **BIT BUCKET** column were voted the most useful by many readers. These letters, hints and kinks, suggestions and even sometimes (not often) gripes are what has directed us in our selection of articles each month.

We have attempted to select subject matter in relation to the percentage of different readers interest. However, it seems that some subjects have more reader interest, than we have received submissions from you. If we are to publish articles on subjects you are interested in, then we need **your input.** Unlike most magazines, 68 MICRO JOURNAL is **YOUR MAGAZINE!** This means that you the reader determines what we publish.

We forfeit thousands of dollars of advertising revenue each year because we screen advertised products to insure that they perform **as advertised.** Some 'bad apples' have slipped through (not many!). However, having tested (we order under an 'assumed' name many items not advertised in 68 MICRO JOURNAL) products advertised in other magazines, we have decided to continue our screening requirements. There are other products that are advertised elsewhere (a very few) that are indeed excellent products. Therefore, it should not be assumed that just because it is not advertised in 68 MICRO JOURNAL it is of questionable quality. I am referring of course to products pertaining to the type computers that 68 MICRO JOURNAL is all about.

So if we are to continue to serve you as we have for the past four years plus, then I need your input on this also. If you have purchased a real 'lemon' let me know. I can assure you we will keep it on file and use the information accordingly. Of all the different groups of computers (Radio Shack, Apple, Commadore, IBM (personal), etc.) we have **less complaints** with our advertisers, than any of the others. We sure do not claim perfection, but I can assure you that you can place more **faith** in advertising in 68 MICRO JOURNAL, than **any** other computer magazine! We have received hundreds of pounds of letters stating as much, from you the readers.

DMW - - -

LABEL PRINTER OS9

CASSETTE LABEL PRINTER

Anyone who makes recordings with a cassette recorder has probably had problems trying to write small enough and clearly enough to fit all the needed information on the small card supplied with the tape.

This program, written in Microware BASIC09 uses an Epson MX-80 printer to print out labels that will fit into the cassette boxes. Each tape has a title that is made up of two lines up to 20 characters long. The title will appear at the spine of the box. Up to 17 selections can be listed for each side of the tape.

There is also space for a tape counter number to help locate each selection on the tape.

After the instructions have been read the program will prompt you for the two line title. If you want to exit the program just type in 'QUIT' in response to the title prompt. After the title is entered you will be prompted for the selections on the first side of the tape. In response to the prompt enter the tape counter number followed by a comma, and then the name of the selection. If you don't know or don't want to use the tape counter number you have two choices. If you just type the comma and the selection name a dash will be placed in the number column. If you type a space followed by the comma and selection name the number column will be left blanc. When you have entered all the selections on the first side type 'OVER' for the selection name and the program will start to prompt for the second side. If you have only recorded on one side of the tape just type 'END' in response to the selection name prompt. Using 'END' when entering the data for the second side will cause the label to be printed. After the printout is complete the program will ask if you want to continue or quit.

The printed labels can then be cut or torn on the dotted lines and folded to fit in the box.

The listed version of the program is heavily commented with remarks. After you load the program it can be packed with the PACK command to save space.

Albert B. Accettola Jr., M.D.
1361 Hylan Blvd.
Staten Island, NY 10305

```
PROCEDURE CASSETTE_PRINTER
 0000     REM A PROGRAM TO PRINT LABELS FOR CASSETTE TAPES USING AN MX-80 PRINTER
 0046     DIM BORDER:BYTE(40)
 0052     DIM NUMBER,PRINTER_PATH:BYTE) B:INTEGER
 0063     DIM NAME:BYTE(31)) TITLE1,TITLE2:BYTE(20)) SELECTION_A(17
          ),SELECTION_B(17):STRING(26)) A(17),B(17),ANSWER:STRING
          (3)
 0084     LET BORDER="----------"
 00E4     LET NAME="/P"
 00EE     OPEN #PRINTER_PATH,NAME:WRITE
 00FA     LET P=PRINTER_PATH
 0104     LET NUMBER=0
 010C     REM INSTRUCTIONS
 0118     PRINT CHR$(12)) CHR$(22)1
 0125     PRINT '     This program is used to print labels that will fit into cassette tape boxes.'
 0178     PRINT
 017C     PRINT '     The title is up to two lines of twenty (20) characters.  Just type in '
 01CA     PRINT 'QUIT on the first line to exit the program.'
 01F9     PRINT '     Enter the tape counter number (3 digit, a comma, and the name of the '
 0249     PRINT 'selection. If you don't know the counter number, entering a space followed by '
 0298     PRINT 'a comma leaves the area blank on the printout. Just entering a comma then'
 02E8     PRINT 'the selection name will cause a ' - ' to be printed in the number column.'
 0335     PRINT 'Entering 'END' for a selection name will stop input and print out the labels. '
 0386     PRINT 'Entering 'OVER' stops input and starts it again for the second side of the tape. '
 03D9     PRINT \ INPUT "DO YOU WANT TO CONTINUE? (Y/N) ",ANSWER
 0408     IF LEFT$(ANSWER,1)="N" THEN 10
 0417     GOTO 999
 041B     REM INITIALIZE ALL VARIABLES
 0436 10  LET TITLE1=""
 0441     LET TITLE2=""
 0449     REM THE NUMBER OF LABELS PRINTED
 0468     LET NUMBER=NUMBER+1
 047C     IF NUMBER>2 THEN PRINT #P,CHR$(12)1
 0AED       LET NUMBER=1
 0493     ENDIF
 0495     FOR N=1 TO 17
 0A45       LET A(N)=""
 0A81       LET B(N)=""
 0A89       LET SELECTION_A(N)=""
 04C9       LET SELECTION_B(N)=""
 04D5     NEXT N
 04E0     PRINT CHR$(11A)) CHR$(22)1  \REM HOME UP AND CLEAR SCREEN
 0505     REM INPUT TITLE
 0513     PRINT '                    ----------'
 0544     INPUT "CASSETTE TITLE - LINE 1 : ",TITLE1
 0583     INPUT '              LINE 2 : ',TITLE2
 05B6     IF T$TLE2="QUIT" THEN 999
 05F9     PRINT
 05F9     REM INPUT SIDE A OF TAPE
 0582     PRINT "SIDE A"
 058C     PRINT '                    NO.    SELECTION NAME'
 05EA     FOR N=1 TO 17
 065A       PRINT "SELECTION NO. "; N) ' - ";
 0617       INPUT '   ',A(N),SELECTION_A(N)
 062E       IF SELECTION_A(N)="END" THEN LET SELECTION_A(N)=""
 0648         LET A(N)=""
 0657         GOTO 200
 0659       ENDIF
 0658       IF SELECTION_A(N)="OVER" THEN LET SELECTION_A(N)=""
 0678         LET A(N)=""
 0687         GOTO 100
 0689       ENDIF
 0689     NEXT N
 0698     REM INPUT SIDE B OF THE TAPE
 0683 100 PRINT
 06B8     PRINT "SIDE B"
 06C2     PRINT '                    NO.    SELECTION NAME'
 06F8     FOR N=1 TO 17
 0700       PRINT "SELECTION NO. "; N; " - ";
 071B       INPUT '   ',B(N),SELECTION_B(N)
 0734       IF SELECTION_B(N)="END" THEN LET SELECTION_B(N)=""
 0751         LET B(N)=""
 0758         GOTO 200
 0761       ENDIF
 0763     NEXT N
 076E     REM OUTPUTS THE LABEL TO THE TERMINAL SCREEN
 0799 200 PRINT CHR$(10)) CHR$(22)1
 07A6     PRINT TITLE1) " - "; TITLE2
 07B5     PRINT
 07B7     PRINT; TAB(10)1 "SIDE A"1 TAB(40)1 "SIDE B"
 07D2     PRINT
 07D4     FOR N=1 TO 17
 07E4       IF SELECTION_A(N)<>"" AND A(N)="" THEN LET A(N)=" - "
 0808       ENDIF
 0809       IF SELECTION_B(N)<>"" AND B(N)="" THEN LET B(N)=" - "
 0834       ENDIF
 0838       IF SELECTION_A(N)="" AND SELECTION_B(N)="" THEN 250
 0852       PRINT A(N)1 TAB(5)1 SELECTION_A(N)1 TAB(35)1 B(N)1 TAB(40
            )1 SELECTION_B(N)
 0878     NEXT N
 0886 250 REM PRINTS THE LABEL ON THE PRINTER
 08A0     PRINT #P,BORDER
 08B6     PRINT #P,CHR$(27)1 "W"1 CHR$(1)1  \REM SET EXPANDED PRINT MODE
 08E4     PRINT #P,TITLE1
 08E5     PRINT #P,TITLE2
 08FA     PRINT #P,CHR$(27)1 "W"1 CHR$(0)1  \REM CANCEL EXPANDED PRINT MODE
 0920     PRINT #P,TAB(7)1 "SIDE A"1 TAB(27)1 "SIDE B"
 094C     PRINT #P,CHR$(15)1  \REM SET CONDENSED CHARACTER PRINT MODE
 097E     PRINT #P,CHR$(27)1 "U"1 CHR$(1)1  \REM UNIDIRECTIONAL PRINTING
 098E     FOR N=1 TO 17
 098C       PRINT #P,TAB(0)1 B(N)1 TAB(5)1 SELECTION_A(N)1 TAB(30)1 B
            (N)1 TAB(43)1 SELECTION_B(N)
 09EF       IF N=12 THEN PRINT #P,CHR$(13)1
 0A06       PRINT #P,BORDER
 0A11       PRINT #P,CHR$(27)1 "W"1 CHR$(1)1
 0A23       PRINT #P,TITLE1
 0A30       PRINT #P,TITLE2
 0A3D       PRINT #P,CHR$(27)1 "W"1 CHR$(0)1
 0A4F       PRINT #P,BORDER
 0A5A       PRINT #P,CHR$(15)1
 0A66       PRINT #P,CHR$(27)1 "U"1 CHR$(1)1
 0A79     ENDIF
 0A7D     NEXT N
 0A86     PRINT #P,CHR$(18)1  \REM RESETS PRINTER TO ORIGINAL STATUS
 0A86     PRINT #P,BORDER
 0AC3     PRINT #P \ PRINT #P \ PRINT #P
 0A86     INPUT "ANOTHER ONE? (Y/N) ",ANSWER
 0AF1     IF LEFT$(ANSWER,1)="Y" THEN 10
 0B04     PRINT #P,CHR$(12)1 CHR$(18)
 0B13 999 END
```

SYMPH. No. 5 & No. 8 BEETHOVEN

SIDE A	SIDE B
Symphony No.5 in C min. Op. 67	Symphony No. 8 in F maj. Op. 93
Vienna Philharmonic Orch.	
Hans Schmidt Isserstedt	
000 1st movement Allegro con brio	000 1st movement Allegro vivace e con brio
- 2nd movement Andante con moto	- 2nd movement Allegretto scherzando
- 3rd movement Scherzo	- 3rd movement Tempo di menuetto
- 4th movement	- 4th movement Allegro vivace

SYMPH. No. 5 & No. 8 BEETHOVEN

Allegro

440 Sonata No. 23 in F min.
Op. 57 Appassionata
Robert Casadesus
end

507 Sonata No.14 in C# min.
Clair de Lune
Robert Casadesus

RUMORS & SUCH

As has been our practice in the past, we try to keep our readers informed on what is likely to happen, in the 68XX world. Sometimes these 'tid-bits' cause 'howls' from vendors who would rather do their own 'announcing'. However, since we publish 68 MICRO JOURNAL for the benefit of our readers it is our responsibility to try to keep you as informed, as we can, and still not do real harm to anyone.

We have respected many request for reasonable delay in publishing information, that might tend to negate the efforts of some manufacturer. For one, those instances that would allow a competing manufacturer to know what his competitor is about to unveil, or has on the drawing boards. That of course would do more harm than good, and be quite unfair on our part. Especially as many are personal friends and often pass information, of a secret nature, during the course of casual conversations. In fact, that has been a real quandary. So if occasionally some bits of interesting information are delayed in getting publishing, you know the reason. However, we still get the jump on about everyone else, and manage to keep you informed.

Possibly by the time this gets into your hands, Radio Shack will have announced the *availability* of MICROWARE'S OS9 disk operating system. My information which comes from a source close to Radio Shack, tells me that it will also be used on other 6809 computers, to be and being developed by Radio Shack. What this means to the 6809 market is somewhat unclear at this time, but it should be a beneficial stroke for other 6809 system manufacturers.

Now that may sound strange. But if you will stop and think, it becomes evident that Radio Shack will, by the very size of their distribution outlets and efforts, make more potential computer users aware of the 6809 and all it has to offer as a CPU, than any other 68XX manufacturer I know of. They have the advantage of sheer market weight.

It is not intended as a slur towards Radio Shack to say that although they produce excellent computers, they do not produce computers as versatile as most existing 6809 computer manufacturers. In fact, it is no secret that Radio Shack sells only a very small portion of the 'add on after-market' items, normally associated with a more complete computer system. Also it is a fact that many potential users came into the small computer market as a direct result of Radio Shack computers. A very large portion of these users changed over to other brands of computers, as their needs exceeded the availability of Radio Shack brand add-ons, or in some instances better and less costly than Radio Shack brands were available. The built-in restrictions, both hardware and software related, of their computers, sent many users afield looking for computers that would allow them expansion and more access to current state-of-the-art add-ons, than available through or for the Radio Shack computers.

In the realm of the 6809 this same market activity will probably exist, and many who come into the small computer user market via Radio Shack products, will in many instances look for bigger and better 6809 machines, as their needs expand. When that time arrives, those existing 6809 computer manufactures that we now know, will have the equipment to **completely** fill these needs. We have observed this already in Color Computer users.

So it is somewhat a paradox, but most 6809 computer manufacturers will actually be aided rather than damaged by the entry of Radio Shack into the 6809 marketplace.

We will be seeing a UNIX™ type system also for the Radio Shack 68000 Model 16 computer. It will come from a well known software house. None too soon either. The present availability of software for the RS 68000 machine is at a very low ebb and can go nowhere but up. However, I expect a rather short life for the Model 16, as it has some rather stiff and unnecessary limitations. It does little to make available the real power of the 68000. It is restrictive in both RAM addressability and add-on capability. These two alone cripple what could have been a fine computer (more on this in a later article).

However, to get back to the point; there will be additional 68XX computers from Radio Shack. If the present 68XX manufacturers keep their R&D aggressive they should reap the fallout.

COLOR COMPUTER INFO EXPANSION
Volunteer Column Editors Needed
And our 'Annual Progress' Report

For the past 6 months or so 68 MICRO JOURNAL has experienced an **explosion** of 'new' Color Computer subscribers. For this we are indeed delighted. Not only for the subscriptions, but because it means that many new **6809** users are 'coming aboard'.

For our advertisers this is good, for it promotes the 6809 CPU, as the finest available! Once a new user discovers what we have had all along, he is a very real potential customer for a larger 6809 system. Hence we all profit. I am told practically daily by some 'older' advertiser that the influx of new 'color computer' users has enhanced their sales. Some of course to a greater degree than others. However, all stand to gain, as it broadens our user base. And lets them know what is available, once they experience the power and ease of using the 6809.

Now we, 68 MICRO JOURNAL, have come to the point that we need to more fully support this new base of readers. We have, in some respects, a different type of user, than we started out with. It seems that the new users, coming in at the color computer level, need more basic information than has been our normal practice in publication. He needs solid information on the more popular disk systems, such as FLEX™, UniFLEX™, OS9™, SSB DOS™, SOOS™ and possibly others. Especially FLEX™, UniFLEX™, and OS9™ which seem to be used by more than 80% of the larger 6809 systems. As some of the other systems become more popular, they should be supported also. This means that **we must have more basic articles** to satisfy this need. Also we will be publishing more 68000 articles, and soon. Meaning that we need volunteer editors to do columns to help us all.

If you feel that you could handle a column every other month or so, **PLEASE DROP ME A LINE.**

I hope that we can soon enlarge the size of 68 MICRO JOURNAL to allow more coverage of the subjects we are now light on. Probably we will have to begin publication of an additional magazine. Similar to 68 MICRO JOURNAL but more fully covering the color computer and other new 6809 computers coming soon.

Let me know your feelings on this and also let me know (soon) if you want to participate. Either way we are going to expand our coverage, but we need your input if it is to succ d, as has 68 MICRO JOURNAL.

As of the end of 1982, 68 MICRO JOURNAL covers more of the 68XX market than all other magazines combined. Advertisers tell me that the only advertising results they see is from their advertising in 68 MICRO JOURNAL. Of course there is a reason, mainly that we have more than **10 TIMES** the subscribers, than any other 68XX magazine. And that is a **FACT!!** That makes the cost of advertising in 68 MICRO JOURNAL about 1/6th to 1/8th of what it is in other 68XX magazines, per reader. And readers is what **COUNTS.** Therefore, our success means a bigger and better magazine for all of us. So please, when you order something from one of our advertisers, tell them you saw it in 68 MICRO JOURNAL.

We will not be 'sitting on our laurels' in the coming year. I can promise you an expanded and more complete 68XX magazine, as we get our expansion program into gear. I tell you this now, however, you already know, 68 MICRO JOURNAL's publishing policies are set by our readers, and the contents of 68 MICRO JOURNAL is furnished by our readers, therefore, as usual, I am just letting you know. Together we will get it done. To all of you our most **HEARTY 'THANKS'!!!**

DMW - - -

QED

QED A do-it-all EPROM BURNER

As time passes it seems, and certainly not a day too soon, that the hardware and software we receive for review looks more and more like **professional** and less and less like hobby. This is a good sign, and looking at all the new systems coming out each month, one wonders where the Standard S50 Bus is headed? Well, I can tell you one thing for sure, it will be around for a long time!

There is just one thing that most of the newer offerings, and a lot of the old ones, can't do; let the user modify the way the 'beast' functions. I have heard too many Apple, Radio Shack, IBM, Cannon, Commodore and users of other brands wring their hands and literally cry because of **built in restrictions.** For some unknown reason I even feel a twitch of remorse when I hear of one of the old (established??) S100 bus manufacturers going down the drain. Actually they had a 'millstone' from day one, namely a bus that had more noise and other problems that was just too much to overcome. However, they did allow the knowledgeable and brave a 'chance' to change the system. The newer stuff just won't let you even try.

Granted that most of the newer users of Standard S50 Bus computers don't do as much 'piddling' as we did years ago (golly, seems like a long time ago) but for those that have the knowledge and skill, the equipment and where-with-all is available. We have that over most all other computer systems. Not only do we have it, but it is mostly **TOP GRADE!**

This review is about such a product. A PROM burner limited only by your skill and equipment. And the rather nice part about this particular 'burner' is that really not much skill is required. It will program any popular (5v) EPROM available; 2508, 2758, 2532, 2732, 2732A, 2516, 2716, 2564 and 2764. All that with a commercial grade EPROM burner that sells for about, well, exactly $125. Yep, $125 and designed to run continually, no wait and cool off states.

UNIQUE Technologies QED EPROM Programmer

This unit fits on the S30 bus of a standard S50 computer. LEDs monitor both the programming and logic voltages during operation. Equipped with a ZIF 28 pin easy insertion socket for both 24 and 28 pin EPROMs. And for maximum safety all pins of the EPROM socket are diode clamped to within one diode drop (appx. .6v) of ground potential.

SOFTWARE

The programmer is driven by menu type software called QED. This software is very user friendly and short of overlaying it with data, or a 'run away' seems difficult to crash. At practically each stage of operation the user may continue or abort, without fear of scrambling the EPROM or other RAM data. Even little things like accepting both upper or lower case characters (most commands to the system are single characters), not having to type the C/R each time, makes things go nicer. Another nice touch is program 'feedback'. One of the most annoying things about using a computer is when you command it to accomplish some function and the cursor either just sits there, or the screen goes blank with the cursor just sitting there (sometimes not) and you

wait and wait and wait.....not really knowing if it is doing it's thing, or suddenly decided to send the 'PC' off on some bit twiddling rampage among the system RAM. A nice thing about this software is that it **tells you** what it is doing, or in other words it issues **progess messages**. Someday I am going to write a manual for those folks who write software. And if I don't get anything else across, I hope that they will design their product so that it always keeps the user informed as to what is happening way down in the bowels of RAM. The old saying, "no news is good news", does not apply when using a computer. I want to know what is happening and what to expect. No data is allowed to roll off the screen until the user is ready. Seems like such a little thing, but to us who still meddle with the 'innerds' of our computer, it makes life a lot easier.

HARDWARE Requirements

The QED system adapts to most 6800 and 6809 systems. The requirements are as follows:

1. A disk system 6800 or 6809 with a 30 pin slot.
2. Disk operating systems FLEX™ or SSB™.
3. An ACIA based terminal or a system that simulates an ACIA based terminal.
4. A CRT or terminal of 24 lines by 80 characters, with data rates of 4800 baud or faster.
5. The display should support home cursor, clear screen and backspace.
6. At least 6K of user RAM plus room in additional RAM for the data to be programmed.

The only real fiddling with the software is getting the 'delay' time correct for your particular system clock. Once done, and assembled, no changes should be required. That is provided you do not change the terminal address (sorta difficult on most systems) or change the QED board to another S30 slot.

Four (4) procedures are given for figuring your system clock **exactly**. While I won't go into them in detail I will tell you that it really isn't too difficult.

If you know that your system clock is exactly 1.0, 1.5, or 2.0 MHz, then the supplied figures will do nicely. However, if your system isn't running at the speed you **believe** it is, then this is a good excuse to check your phase 2 clock.

Method 1. Values are given for the speeds above.

2. Approximation: Here all you need is a digital watch or clock. Just assemble and run the furnished program 'SIGHT'. This method will get you within 2% of your actual system clock speed, and should do fine as most manufacturers require an accuracy of 10%.

3. Calculation: this method requires a frequency counter. However, it will get you closer than method 2, but probably won't do any better.

4. Direct: This method requires an oscilloscope. While this method will get about as close as method 3, again it is probably not any better than method 2.

PROGRAMMER Software

The furnished software, in source code, is called QED9, at least it is for the FLEX™ (6809) version we are reviewing. It may be 'ORGed' at any place in RAM you might desire and is relocatable (6809 only). One thing we were **really** impressed with is that **it is completely commented, completely!** This to us is very, very important. Too much supposedly 'commented' software we see is lacking in areas that apparently the author felt were so simple that that particular portion of code needed no comments. Some programmers are so proficient that they loose sight of the simple fact that not all users understand it as well as they do. I feel that I write fairly clean code, however, there have been times while modifying code that was supposed to be 'completely' commented, that I had to stop and do some paper bit twiddling to figure what to do next. For some that can spell disaster. These source programs are a **joy** to read. Just like a good book (right, Mickey??), or was it newspaper?

The necessary information needed to be placed in the source before assembly by the user is as follows:

The address of the S30 slot where the board will be positioned.

The address of your terminal ACIA. (Video boards that simulate an ACIA such as the FEBE's VIDEO PORT can also be used, PIA driven video boards will require user interface).

The backspace character for your system, this is by standard the code **08**, but on some systems beware, it may be different. If it is you can insert your backspace character.

The address of your monitors warmstart routine. Instructions are included on how to determine most monitor warmstart addresses.

Also you will be required to furnish the warmstart address for your disk operating system.

The FMS CLOSE routine needs to be inserted.

The 'Utility Command' starting address is needed.

The 'delay' mentioned above needs to be inserted in the source code also.

The number of lines on your CRT or terminal is needed so that messages may be not only centered but not scrolled off before you are ready.

The address where you desire to place the data to be programmed.

The final information needed prior to assembly is the characters needed to cause your CRT or terminal to do a 'home up and clear screen' function.

The above may seem complicated, but in practice it is simple and took about 15 minutes of looking up values and addresses in our 'FLEX users manual'. SSB users should have it as easy.

OPERATION

Operation is simple due to the interaction from the menu. The first menu is the 'MASTER MENU'. Here selection is allowed for calling any of the following actions required for the particular burning session.

Load buffer with a constant. Examine or change buffer. Load buffer from memory, load buffer from EPROM, relocate this program, execute other disk commands (must not overlay main program or data buffer), alter buffer origin, hex dump of buffer, verify EPROM programmed, verify EPROM erased, select EPROM type (used to burn an EPROM different from the one originally specified while assembling the furnished source code), program EPROM, exit to the monitor or exit to the disk operating system.

The Master Menu also informs the user of what type EPROM is expected by the system, as called. Also the data buffer is shown from starting to ending address, along with a copyright blurb, version numbers, etc.

DOCUMENTATION

The documentation furnished is complete, even to including a sample programming session. Each of the menu selections is covered in detail. The average user will experience little if any difficulty using this system. The 'advanced' portion (you know, back a couple hundred words, where I told you about the information you furnish the source code prior to assembly) is detailed to allow a completely custom version for your particular requirement. Most EPROMs and other changes can be declared at 'burn' time, but for our needs a custom program is better. This way we just drop in a clean EPROM, verify that it is clean and let-'er-rip.

The BOARD

The S30 slot board is of high grade epoxy glass. It is solder plated and silk screened with each component marked. All ICs are socketed and ours came with gold plated bus pins which are standard.

The Personality Module

For each type of EPROM, 2708-2564 (16K 2528 and 27182 types will also be supported), there is a 'Personality Module' mated into a 16 DIP IC socket. This unit straps the EPROM socket to a portion of the main circuit board, allowing the different types to be programmed. It is a simple matter to make your own (instructions included in manual). Normally furnished module is a 1K and 2K module.

Conclusion

This board is ideal for both occasional use and heavy duty applications. The system requires no 'cool down' period between EPROMs and does not unduly load the system power supply. However, users of older systems with light power supplies might want to insure that their system, if 'loaded' with other boards, is capable of maintaining adequate power to the rest of the computer. Special attention should be given to the +8v (or so) source supply, as this is the one that normally loads down first. This applies when any additional board is added to the system.

Also I was told by Richard Rae, of UNITEK that should any customer experience problems configuring the system, that they would be willing to configure the system **at no charge**, provided the customer could supply the necessary information. Most standard systems would require little knowledge on the customers part. Which

makes for a super package, and the price is right.

Additional information or to order contact:

UNITEX
PO Box 671
Emporia, VA 23847

A 68 MICRO JOURNAL Review - - -

PB4-BUFFER

PB4 Intelligent Port Buffer

A few months back we did a review of a new PROM/RAM board by **ACORN Computer Systems**. The board was a new and novel addition to the Standard S50 Bus systems. At that time they promised some additional 'new and novel' boards for our bus. Well, here is another.

The PB4 is an intelligent port buffer. What this essentially means is that it does its own 'thinking' and also stands between the computer and whatever it is attached to. In our case a parallel dot matrix printer (MX80).

It should not be confused with a spooler. FLEX™ already has a spooler. A spooler is a multi-t sking program that allows, in the case of FLEX, a printer to be printing out a file or listing, while the user regains control of the keyboard and computer, and continues to do other tasks. This has the restriction that no other printing task can be started while the spooler is running.

Now a 'Port Buffer' is another thing. A port buffer absorbs speed differences between two objects, our case a computer and a printer, and by storing all or most of the task to be buffered, and also having its own thinking system (in this case a 6802 CPU), along with necessary coupling devices (our case PIAs) and other associated digital devices, it drives the printer and the operator regains control of the keyboard and computer. So like spooling we can do another task while the printer is printing out a file or listing, etc.

The primary advantage over a spooler is that the port buffer allows things to run faster and without delays and interrupts to the disk system. There are two delays that the PB4 will eliminate. First, the printer is much slower than the computer. Therefore, keeping the printer running during disk operations and keyboard inputs eliminates one. econd, During data output the computer outputs lines of characters, a carriage return and a line feed. The printer will accept characters up to the carriage return. Upon receiving a return this line feed is then printed. The printer will not acknowledge this line feed or release the printer until the line is completely printed. The 'port buffer' having stored these control characters, as received, continues to drive the printer, from its memory, thereby releasing the computer for other task.

The PB4 also entails two extra buffered control lines at an output header. These lines can control the printer, or any other device to be driven by the on-board PIAs.

There are two versions of this board. One, like ours, is used as a buffer. However, it can function as a 'stand alone' 6802 computer. It comes with 4K of RAM, but instructions are included to allow expansion to 8K.

It comes as bare board, kit or wired and assembled.

There are extra pads on the board. These allow strapping the board for all those functions necessary in single board computer operation, including master reset, etc. Basically it consist of a 6802 CPU, 4K (expandable to 8K) of RAM, a 2K monitor and 28 I/O lines.

A universal PB4 is also available, in kit or assembled form. This board trades one of the PIAs and the Molex connectors for a reset timer circuit and a 24 pin input header. The control PROM (monitor) is a 2716(5v) EPROM and the RAMS are 6116 types.

Documentation

As usual the documentation is good. It even includes a section on proper soldering (ala Heath). Also included is complete construction instructions and drawings and circuits. A battery backup option is explained for those needing this feature. A real 'bug' to many who construct kits, where 'hankshaking' is used, is the lack of much information concerning handshaking. This is covered. In fact, to adequately describe the documentation, it would be practically necessary to reproduce the entire manual. Needless to say, but the more they tell us, the easier it is. This board should be no problem to the average kit builde type.

Software furnished

Because most users will use the PB4 to drive a printer they have included source listing (commented) fo Epson type printers, including the 'hard reset' required by earlier Epsons. Also included is a source program, in 6800 that is 'Centronics' compatible. This program is the board driver routines. Also it does a NMI self test and a RAM test and report to the printer, for the PB4 board.

How it works

As stated earlier, the PB4 is a 'Single Board Computer' in its basic configuration. The 6802 uses the normal Motorola type mnemonics and has 128 bytes of RAM at $0000 to $007F. A battery backup option is provided for the first 32 bytes of RAM. The RAM sockets may be jumpered as either all RAM or EPROM. By piggy-backing the memory, you can have a total of 10K or PROM or 8K of RAM and 2K or PROM.

Twenty-eight (28) I/O lines (input/output) are provided. This consist of 8 buffered input lines, 8 UNbuffered programmable I/O line, 8 buffered output lines and 4 buffered control lines. The control lines are 2 output only, 1 input only and 1 programmable. All lines are non-inverting.

There are 2 PIAs. One PIA is the interface between the 6802 CPU and the S30 bus. It is not used in the stand-alone mode. A 74LS244 is the input bus driver. It is a tri-state device and is controlled by the 6802 CPU. The other PIA is the output device with its lines buffered by 2 8T97s. Handshaking is the standard Centronic type and is accomplished by CA1 and CA2 of the PIA.

Input hankshaking is somewhat more complex. CA2 on one PIA is connected to CB1 on the other PIA. When your computer sends a "data sent" to the board an IRQ is done by the 6802 CPU. The CPU toggles the other PIA when the IRQ is output by the CPU. This allows complete handshaking between the board and the computer. By this method the 6802 CPU continues to service the output device, normally a parallel printer.

Conclusion

The PB4 is constructed of 2 oz. glass-epoxy copper board, double sided with plated through holes and solder masked.

For the user who 'spools' large amount of data to a printer or other device, this system can save a lot of time. Also I had stopped using the spooler in FLEX due to the starting and stopping of the printer and the disk system, as the parallel task accessed the disk system. This causes more distraction, at least to me, than normal spooling is worth. We do more text editing on some of our 6809 systems than any other, or all other, task combined. Being able to output to the printer and not having the printer stopping and starting, is well worth the price of the system. e s that every time the printer stops, due to a disk access normally, I stop also, loosing time and most often forgetting what I was about to type next. The PB4 board just lets the printer hum on and on, never pausing or stopping.

A note from Merle Giesfeldt tells me that they will also be coming out with an "Extended Addressing System". It is claimed to transfer 20K in .25 seconds. To go along with this system will be a 168K PROM board and a 2764 PROM burner. Will review this system sometim s in the near future.

For Additional information contact:

ACORN Computer Systems
11931 W. Bluemound Road
Wauwatosa, Wisconsin 53226
(414) 257-0300

For current prices of kits and assembled boards see advertising, 68 MICRO JOURNAL.

A 68 MICRO JOURNAL Review - - -

AUTO-COMM

AUTO-COMM A "SMART" Modem Program & Hardware

One of the more popular additions, to the computer both at home and at work, has been a telephone modem

and appropriate driver software. Here at CPI (Computer Publishing Center), headquarters for 68 MICRO JOURNAL, we have installed a modem system, using the Thomas Instrumentation **MODEM BOARD**. It has been on line for a few months now and already hundreds of readers (and others) have used the system, passing both messages, suggestions, complaints (not many), source (text) files and downloading to their systems source files from our modem system. We will soon install another 6809 system to handle files to be downloaded from authors and Associated Editors from their remote (out-of-town) sites. We will be using the system described in this review. See **SYSTEMS designware** advertising.

OVERVIEW

Auto-Comm automates many of the tedious aspects of working with a time sharing main frame computer.

1. Automatic dial-up of other computer systems.
2. Copy files from other computers to your system and store on your disk system, if desired.
3. Send a file from your system to another system.
4. Send "special" or frequently used messages automatically (auto-sign-on, etc.).
5. Conveniently change communications formats (UART formats, etc.) and control echo/no echo.
6. Print during a run, or later.
7. Send electronic mail to other computers, provided they are equipped with Auto-Comm or other similar program.

Features

25 commands, conveniently accessed through 'HELP' functions. An easy-to-use single mode system, with friendly command interpreter.

System Hard/software Requirements

Auto-Comm runs on a 6809 computer under the FLEX™ disk system. It requires 16K or more of available RAM. And a serial port such as the SWTPC S-2 Serial Card.

Also required, of course, is a modem to connect to the telephone lines.

A 'pulse type' dialer is required if 'auto-dialing' is desire. The documentation includes complete instructions for constructing a pulse dialer, consisting of 4 transistors, 2 opti-isolators, 1 zener diode and 4 power diodes, standard (type 1N4002 or similar). Of course if you do not need the 'auto-dial' feature, then the dialer can be forgotten.

Also to use the 'auto-dialer' 2 parallel lines are required, such as from a PIA.

Auto-Comm was originally designed for the SWTPC S/9 6809 computer systems with the MPID board and a S-2 serial port, otherwise a parallel port must be made available and the software patched to reflect the address of the parallel port. Complete instructions are provided and are little cause for concern for most users.

Using AUTO-COMM

In normal operation Auto-Comm is quite simple. Little reference to the manual is needed as Auto-Comm has a 'HELP' mode and also prompts the users in such manner that operation is hassle free.

All commands are organized as a 'tree structure' and the 'HELP' command is the root, the secondary HELP commands are intermediate nodes, and the functional commands are the leaves. Although this may seem complicated, in actual practice it is of **no** concern to the user. It does however, from the efficiency aspect of the binary code (program) help to insure fast and error free operation.

The command (,HELP), all commands start with the comma (,), will produce the following screen display of secondary commands:

```
,MSG
,SCREEN
,CALL
,FILES
,FORMAT
,EXIT
```

Typing a command will list all commands within this group, along with a brief description of each.

Typing (,SCREEN) will display all 'display' commands, and typing (,CALL) will display all 'dialer' related commands. The list of commands is extensive, but not overpowering. The user soon feels right at ease with the command structure and will appreciate the 'user friendly' aspect of the system.

Command ,HELP prints the command group, as indicated above. Command ,MSG prints the following message related commands:

```
,LISTM  Display your prepared messages.
,SENDM  Transmits a prepared message.
,Sn     Transmit message n.
,LOADM  Load message from a file.
,SAVMSG Save messages and phone numbers on disk.
,SETMSG Enter a message into a message buffer.
,BREAK  Send a one second break.
```

Command: ,SCREEN

```
,ECHO   Turns screen echo on.
,NOECHO Turns screen echo off.
,ERASE  Erase all text buffers.
,PRINT  Turn printer on.
,NOPRINT Turn printer off.
,LF     Add line-feed after receiving C/R.
,NOLF   No line-feed after receiving C/R.
```

Command: ,CALL

```
,LISTPHN  List all recorded (saved) phone numbers.
,LOADPHN  Load phone numbers and messages from disk.
,DIAL     Dial a recorded phone number.
,SETPHN   Enter a new phone number in a phone buffer.
,SAVPHN   Save phone numbers and messages on disk.
,HANG     Disconnect from phone line (hangup).
```

Command: ,FILES

```
,SAVETX  Save text file to disk.
,SENDF   Send a disk file through modem.
,CLSAVE  Close disk save file (abort).
,CLSEND  Close disk sending file (abort).
```

Command: ,FORM

Prints a list of UART data formats (8 allowed).

Command: ,EXIT

List the exit command.

Special Commands:

The ,FORMAT command allows the user to change communications formats (7 or 8 bits, even or odd parity, local keyboard echo and other necessary data). ,FORMAT is entered and a special FORMAT menu is put to the screen, the group is listed and selection is done by number. ,FORMAT is so structured so as to make the accidental destruction of good format data difficult.

Program Structure

Auto-Comm loads to low memory and uses all memory up to the 'End of Memory' pointer in FLEX™. The 'text buffer' is set up as a circular buffer. Different pointers maintain buffer data such that it appears as a separate print buffer (hard copy), and another for saving data to a named text file for disk storage.

Auto-Comm requires 8K of RAM and uses an additional 4K for messages and I/O buffers. The remainder of free RAM is devoted for text file storage. Data received but not needing to be saved can be flushed from the buffer by the command ,ERASE.

Received data that is to be saved to disk is handled by the ,SAVETX command. The sending computer must respond to XON/XOFF commands. If communicating with a computer that uses non-standard XON/XOFF symbols or commands, the change may be made, to Auto-Comm as necessary. If the XON/XOFF characters are different between Auto-Comm and the other computer, or the other computer does not use any type of XON/XOFF, Auto-Comm will detect this and limit incoming files to text buffer length.

Binary files may be exchanged, however, they must be converted to hexadecimal code. This allows XON/XOFF to function.

Auto-Comm documentation includes complete instructions, in addition to those previously mentioned to accomplish software or hardware modifications as required by most non-standard 6809 systems.

Conclusions

Auto-Comm is a very user-friendly system. Designed for the SWTPC S/9 series of computers, it adapts easily to practically any other 6809 computer having the necessary port functions as mentioned above. Like any other serious software it looks more complicated than it actually is in practice.

For those desiring a complete modem program then Auto-Comm is a must. The documentation is complete and the necessary hardware (pulse dialer) is simple to construct (pre-wired dialers may be available soon) and install.

For additional information or to order contact:

SYSTEMS designware
6712 E. Presido St.
Scottsdale, AZ 85254
A/C 800 272-4817 - AZ 602 991-1657

A 68 MICRO JOURNAL Lab Review - - -

PERFECT NUMBER

```
Name:   INTP6809.TXT
Purpose: Various interpreters for the 6809

James C. Brookfield             Oct. 11, 1982
1803 Cayuga
San Antonio, Texas  78220
512-434-2072 at home.           512-533-1228 at work
Dear Mr. Williams,

Enclosed are several interpreters for use on the 6809. Several have been implemented and tried out
this "toy" forth for 8C, 12C, 14C, and 16C-infinity interpreters. I now or far 70 use 16C-infinity
interpreter for most of my programming (it is the most compact and most general).

All these interpreters have a common stack interface or discipline. By making all major subroutines
use this interface, debugging and coding go much easier (i.e. subroutine calls are consistent).

This effort is primarily experimental and being pursued as a hobby. As a programmer and engineer
with several years experience, I find the 6809 better than the PDP11's and alot cheaper than the VAX
for this kind of work (PDP & VAX trademarks of Digital Equipment Corporation).

For a more theoretical treatment see pp31-34 of the June 1982 issue of Computer Architecture News
published by ACM.

Also included on the floppy is LARGPRIM.TXT which will generate all the Mersenn perfect numbers or all
the known Mersenne primes. LARGPRIM uses the 7C interpreter. (I is not the most efficient, I
believe a scaling factor of 8 is possible.

There are three routines that must be implemented for each type of interpreter. NEXT is the code
following every machine code subroutine which goes up one level in stack and executes the next
machine code subroutine. DSD is the code preceding every compressed code interpreter string which
selects the appropriate interpreter and causes the following string to be interpreted. Each such
string must end with an ASD code which is interpreted to cause return to the string one level
higher. DSD is named for "descend" or "call". ASD is named for "ascend", and NEXT comes from forth
for "next".

    U is the value stack pointer
    S is the return stack pointer
    Y is the code scanner
    X & D can be used within subroutines, but are sometimes used
      by the interpreters between subroutines. This forces all
      parameters to reside on the value stack and results returned
      on same.
    Direct page is set to the page containing the DSD and ASD routines.
      This is an optimization to allow direct page jumps to several very
      common routines.

    For 8C, 10C, and 12C a "JMP DSD" is the first word of the 8C, 10C, or 12C code string. The code
for ASD is the last code of a 8C, 10C, or 12C string.

    Threaded code(TC)  --often used in Fortran's DSD  PSHS Y      save code scanner
      LDX  -2,Y  get adr of the JMP DSD
      LEAY 2,X   advance past the JMP DSD and place in Y
      next

    NEXT is short enough to be used inline in all machine language
    routines(most used MACRO
      JMP [,Y++]
      ENDM
ASD   PULS Y  the address of ASD is the last code of a TC string
      next


    Indirect threaded code(ITC)  --often used in Forth DSD  PSHS Y
      LDX  ,X
      LEAY 2,X
      next

    also short enough to be used inline next MACRO
      LDX  ,Y++
      JMP [,X]  X has adr of symbol entry area
      ENDM
ASD   PULS Y  the adr of the adr of this ASD is the last code of a ITC
      next    string


    Compressed threaded code(CTC)  --often used in Pascals and Basic's DSD  PSHS Y
      LDX  ,X
      LEAY 2,X
      next
next  MACRO
      JMP [,X]
      ENDM

    Y-reg is not so short in this code, so the code is in the direct
    page where it can be reached easily  NOT  LDX  OPTBL  address of jump table
      LDB  ,Y+
      ABX
      ABX
      JMP [,X]
ASD   PULS Y  the jump table must have an entry for this routine
      next

    Machine code(MC)  --the traditional route DSD  Comp with the JSR or BSR last version
next  NOP  threaded
ASD   RTS

        What all of this implies is that a coding standard is possible for the 6809 which will allow
the user his choice of interpreters, while sharing subroutines and code with users of other
interpreters. The standard or discipline is:

  All arguments and results are passed on the U stack.
  All return addresses are on the S stack.
  The Y register is used as the code scanner.
  The D & X registers are not preserved between subroutines.
  X is sometimes stuffed with an address by NEXT.
  The DP register is dedicated as the direct page of the interpreter.
  All subroutines must have a unique name.
  Subroutines which behave identically but use different subroutines
  will be differentiated by a unique suffix. MC code name suffix will
  identify the make and model of microprocessor (in case the standard
  is applied to another micro or the instruction set of the 6809 is
  changed).
  Interrupts are allowed.
  The required operands and produced results should be documented,
  the type declarations of Pascal being a good model.
  A short description of the subroutine should be furnished.
 A subroutine will consist of either machine code obeying the above
 restrictions or a list of subroutine names. The first name of
 the list must be JMP DSD, the last must be ASD. ASD will have the
 zero code if possible.
 Local symbols if required are presumably pushed onto the U stack as
 encountered. An exception is loop indices which are preferably placed
 on the S stack (with care).
 When nested local symbols are desired a frame pointer is used. On
 entry to the routine the previous frame pointer is pushed on the
 value stack and the new frame pointer set to the current value of
 the value stack pointer. On exit the previous value of the frame
 pointer is restored. The frame pointer resides on a global symbol
 table entry.
 In interactive applications it is expected that a symbol table
 will be used. As a departure from forth, it is required that an
 entry be allowed to have two semantic values. This allows a variable
 both a load and a store routine.

  A symbol string (in front less length).
  A next symbol link.
  A one word code describing the type of symbol.
  The second or alternate semantic pointer (if needed).
  The first or normal semantic pointer.
  A value field (any length).


        The routines to reference variables in the symbol table varies with each interpreter. The
code for 16 bit loads and stores is:

        16C: The normal symbol semantic entry contains a JMP to a LD. The alternate set Y contains a
JMP to a ST. The load and store routines must be in the direct page so a jump to them takes but one
word (two bytes).
v_LD  LDX  -2,Y
      LDD  2,X   the symbol value follows the normal entry
      PSHU D
      next
v_ST  LDX  -2,Y
      PULU D
      STD  4,X   the alternate entry proceeds the normal entry
      next

        12C: The normal symbol semantic entry contains the address of v_LD, the alternate entry the
address of v_ST.
v_LD  LDD  2,X
      PSHU D
      next
v_ST  PULU D
      STD  4,X
      next

        10C: Following the jump table is the value table with at least two bytes per entry, so that a
combined load & store have four bytes of value to work with. The load entry is the address of v_LD.
This implies 256 words for the jump table followed by 256 words for value entries. Each symbol has
its own entries in the jump table.
v_LD  LDU  512,X
      PSHU D
      next
v_ST  PULU D
      STD  512,X   assuming the store entry proceeds the load entry
      next

        Some additional interpreters

        7C-11C  hybrids, based on some property of a code list address, so that address is used either
as the address of MC or the address of the address of MC.

next  LDX  ,Y++  17 & 20 memory cycles
      BPL  nxtb  X always modified
      JMP  [,X]  uses sign of adr (ie all MC is high memory)
nxtb  JMP  [,X]

nxt   LDD  10nt  21 & 26 cycles
      ORB  ,Y    X only changed on ITC
      BNE  nxtb  uses list value*256 (ie all MC at > 1st*256)
      JMP  1,Y++)
nxtb  LDX  ,Y++
      JMP  [,X]

        CTC-11C hybrids, based on some property of the byte code, so that the byte is used either as
the index of a jump table or as half of the address of the address of MC.
(note that variables need not be associated with the jump table.
```

```
nxt LDB  ,Y+   24 & 29 cycles
    BPL  nxta  X always changed
    LDX  #0!   uses sign of code (all pointers in low memory)
    ASLB
    JMP  [B,X]
nxta LEAY 1,Y
    LSA  -2,Y
    JMP  B,X]

nxt LDB  ,Y+   24 & 33 cycles
    BITB !     X always changed
    BEQ  nxta  wmpe for bit code(all indirec  adrs at even locations)
    LDX  #0!+127
    JMP  [B,X]
nxta LDA  ,Y+
    TFR  D,X
    JMP  B,X]

nxt LDB  !0nt  32 & 26 cycle
    CMPB ,Y    X only changed on ITC
    BHS  nxt   uses limit value. Int must be >127
    LDB  ,Y+
    ASLB
    STB  nxt0+3
nxta JMP  !rb!!
nxtb LDA  ,Y++
    JMP  [,X]

 TC infinities, these follow indirect address chain until adr of some particular type found.
nxt LDX  ,Y++  17, 29, 41, ... cycles
    BPL  nxta  X always changed
    JMP  ,X    uses sign (all MC in upper half of memory)
nxta CMPB ,X
    BHI  nxtc
nxtb LDX  ,X   indirection loop
    CMPB ,X
    BPL  nxtb
nxtc JMP  B,X]

nxt LDA  !0nt  21, 33, 45, ... cycles
    CMPA ,Y    X only changed if indirection takes place
    BHS  nxta  uses limit value*256 (all MC above int=256)
    JMP  [,Y++]
nxta LDX  ,Y++
    CMPA ,X
    BLO  nxtc
nxtb LDX  ,X   indirection loop
    CMPA ,X
    BHS  nxtb
nxtc JMP  B,X]

 CTC=ITC infinities, indirec s until code of specific type found.  When found, used  to  index  jump
table.

nxt LDB  ,Y+   20, 44, 56, 68, ... cycles
    BNE  nxtb  X only changed on indirection
    LEAY 1,Y   and left with adr of code
    LDX  -2,Y  uses sign of code (all MC in high memory)
    LDB  ,X
    BMI  nxtb
nxta LDX  ,X   indirection loop
    LDB  ,X
    BPL  nxta

nxt LDB  ,Y+   36, 48, 65, 78, ... cycles
    BITB !     X only changed on indirection
    BNE  nxtb  and left with adr of code
    LDA  ,Y+   uses even/odd code
    TFR  D,X   in code last for byte of adr
    LDB  ,X    proceeds high byte of adr
    BITB !
    BNE  bxtb
nxta LDA  -1,X  indirection loop
    LDB  ,X
    BITB !
    BEQ  nxta
nxtb STB  nxtc+3
nxtc JMP  [!bb!]

nxt LDB  !0nt  32, 65, 52, 69, ... cycles
    CMPB ,Y    X only changed on indirection
    BHS  nxta  uses limit value. Int must be >127
    LDB  ,Y+
    ASLB
    STB  nxter3  stuff jump adr
nxta JMP  !rb!!  !bl at one gives .irect page boundary
nxtb LDX  ,Y++
    CMPB ,X
    BLO  nxtd
nxtc LDB  ,X   indirection loop
    CMPB ,X
    BHS  nxPc
nxtd LDB  ,X
    ASLB
    STB  nxter3
nxta JMP  !jb!!
```

The RTS interpreter Y and S registers are exchanged. S is the code scanner, hence interupts are not possible as they would corrupt the code. Y is the control stack.

```
Next MACRO
     RTS
     ENDM

DEC  STS  ,--Y
     LDX  -2,S
     LEAS 2,X
     Next

ASD  LDS  ,Y++
     Next

     Virtual address space interpreters
```

An address space the size of disk is chosen. Any of the above interpreters may be adapted to the larger address space. However typically a CTC-infinity code would be chosen which allows variable length address codes.

```
0Dxxxxxx  addresses 0..63 mean push 0..63 onto the value stack
01xxxxxx  addresses 64..127 mean do built-in operators 64..127
10xxxxxx  is followed by 8 more bits of address (14 total)
11Oxxxxx  is followed by 16 more bits of address (29 total)
111xxxxx  is followed by 24 more bits of address (29 total)
```

Timings

Type	NEXT	DEC	ASD
MC	13(13R+A#S)	0	0
RTS	5(RTS)	20+Next	9
TC	9	19+Next	7
ITC	16	18+Next	7
CTC	24	18+Next	7
TC-ITC	17/20 (using sign bit)	18+Next	7
TC-CTC	21/28 (using limit)	18+Next	7
TC-Int	17,29,41.. (using sign bit)	18+Next	7
TC-Int	21,33,45.. (using limit)	18+Next	7
CTC-ITC	24/29 (using sign bit)	18+Next	7
CTC-ITC	24/33 (using low bit)	18+Next	7
CTC-ITC	32/26 (using limit)	18+Next	7
CTC-Int	20,44,56,68.. (using sign bit)	18+Next	7
CTC-Int	20,48,63,78.. (using low bit)	18+Next	7
CTC-Int	32,45,57,69.. (using limit)	18+Next	7

Note that the RTS and TC interpreters are faster than MC. This means that if your program consists of a sequence of subroutine calls (nowdays considered good programming practice), then either RTS or TC will be faster than the JSR to the subroutine and the associated RTS back. In addition the code will be more compact. This leads me to conclude that only the lowest level subroutines should be coded in machine code (MC). All other should be coded in interpreter code.

```
*  Name:   LARGPRIM.TXT
*  Purpose:  List Mersenne primes
*   and primes of form 3*2**n-1 & 9*2**n-1
*   and primes of form 3*2**n+1, 5*2**n+1), 7*2**n+1 & 9*2**n+1
*   and the known perfect numbers
*    which for Mersenne prime 2**n-1
*    equals (2**n-1)*2**(n-1)

*  Proper set of constants selected by commenting out the others

*  Numbers listed without CR-LF, so CRT must have wrap-around to
* see complete numbers.  This program takes 55 minutes!  Normally
* O.CMD used to save results.  File generated will be up to 160
* blocks long.  Perfect numbers take 0.4 hours and 331 blocks.
* (1Mhz 6809 system)

*  Flex adrs
OUTCH  EQU $CD0F
PCRLF  EQU $CD24
INCH   EQU $CD0C
OUTDEC EQU $CD39  B=0, adr of num in X
FLEX   EQU $CD03

*  Interpreter
NEXT MACRO .
 JMP [,Y++]
 ENDM

START LDU #PRMS
 LDY #PROG
 NEXT

*  Code for primes
#PROG FDB CRLF,CLEAR,POOL,MISC
* FDB POOL,DBL,LOOP
* FDB HEADNG,CRLF,DEC,PRTNUM,CRLF,CRLF,LOOP,FLEX

*  Code for Perfect numbers
PROG FDB CRLF,POOL,MISC2
 FDB CLEAR,POOL,DBL,LOOP,DEC         form 2**n-1
 FDB POOL,DBL,LOOP              and double n-1 times
 FDB HEADNG,CRLF,PRTNUM,CRLF,CRLF,LOOP,FLEX

*  Routines

*  Set number to 1
CLEAR LDX #NUM
CLL CLR ,X+
 CMPX #NUMEND-1
 BNE CLL
 LDA #1
* LDA #3
* LDA #9
 STA ,X
 NEXT
*  Double number (base 10)
*  Optimized by unwinding & expanding length num
DBL LDX #NUMEND
 PSHS Y
 CLC
DBLL LDA ,-X
 ADCA ,X
 DAA
 STA ,X
 LDA ,-X  do 4 digits per pass
 ADCA ,X
 DAA
 STA ,X
DBLA LEAY 2-NUMEND,X  offset modified as size increases
 BNE DBLL
 BCC DBLB  expand size if carry
 INC ,-X
```

```
 LDD DBLA+2
 ADDD #2
 STD DBLA+2
DBLB PULS Y
 NEXT

* Subtract one from number (with no borrow)
DEC DEC NUMEND-1  for k*2**n-1 primes
*DEC INC NUMEND-1  for k*2**n+1 primes
 NEXT

* Print number (2 bcd digits per byte)
PRTNUM LDX #NUM
PRTLA TST ,X+
 BEQ PRTLA
 LDA ,-X
 ANDA #$F0
 BEQ PRTB
PRTL LDA ,X
 LSRA
 LSRA
 LSRA
 LSRA
 ADDA #'0
 JSR OUTCH
PRTB LDA ,X+
 ANDA #$0F
 ADDA #'0
 JSR OUTCH
 CMPX #NUMEND
 BNE PRTL
 INC NUMEND-1   for k*2**n-1 primes
* DEC NUMEND-1   for k*2**n+1 primes
 NEXT

* Enter count-down loop
POOL PULU D
 PSHS D,Y
 NEXT

* Exit test on count-down loop
LOOP PULS X
 LEAX -1,X
 PSHS X
 BNE LOOPN
LOOPA LEAS 4,S
 NEXT
LOOPN LDY 2,S
 NEXT
* Prepare loop count (0 times to double number)
* Value stk= last prime, this prime,...-> difference, this prime,...
MISC LDD 2,U
 SUBD ,U
 STD ,U
 NEXT

* Prepare loop counts for perfect numbers
* Value stk= last prime, this prime,...->this prime, this prime-1,
*        this prime,...
MISC2 LDD 2,U
 PSHU D
 SUBD #1
 STD 2,U
 NEXT

* Print heading
* Value stk= this prime, ...
HEADNG LDX #MSG1
 JSR PSTRNG
 TFR U,X
 CLRB
 JSR OUTDEC
 LDX #MSG2
 JSR PSTRNG
 NEXT
* Header for primes, select one MSG1 and one MSG2
#MSG1 FCC ' 2 ** ',0
*MSG1 FCC ' 3 * 2 ** ',0
*MSG1 FCC ' 5 * 2 ** ',0
*MSG1 FCC ' 7 * 2 ** ',0
*MSG1 FCC ' 9 * 2 ** ',0
MSG1 FCC 'Perfect number for n=',0
*MSG2 FCC ' - 1 =',0
*MSG2 FCC ' + 1 =',0
MSG2 FCC ', (2**n-1)*2**(n-1) =',0

* Print string, not threaded
PSTL JSR OUTCH
PSTRNG LDA ,X+
 BNE PSTL
 RTS

* Print CR-LF
CRLF JSR PCRLF
 NEXT

* These constants are initialization for value stack, which is # of
* exponents, zero & the exponents.

* Primes of the form K * 2 ** N - 1
PRMS  FDB 26  # of primes, K=1,  Mersenne primes
      FDB 0,2,3,5,7,13,17,19,31,61,89,107,127,521,607,1279,2203,2281
      FDB 3217,4253,4423,9689,9941,11213,19937,21701,44497
*PRMS  FDB 31  K=3
*      FDB 0,1,2,3,4,6,7,11,18,34,38,43,55,64,76,94,103,143,206,216,306
*      FDB 324,391,458,478,827,1274,3276,4204,5134,7559,12676
*PRMS  FDB 28  K=9
*      FDB 0,1,3,7,13,15,21,43,63,99,109,159,211,309,343,415,469,701
*      FDB 871,939,1551,3115,3349,5589,5815,5893,7939,8007,11547

* Primes of form K * 2 ** N + 1
*PRMS  FDB 5  K=1
*      FDB 0,1,2,4,8,16
*PRMS  FDB 24 k=3
*      FDB 0,1,2,5,6,8,12,18,30,36,41,66,189,201,209,276,353,408,438
*      FDB 534,2208,2816,3168,3189,3912
*PRMS  FDB 16 K=5
*      FDB 0,1,3,5,7,13,15,25,39,55,75,85,127,1947,3313,4687,5947
*PRMS  FDB 22 k=7
*      FDB 0,2,4,6,14,20,26,50,52,92,120,174,180,190,296,320,390,432
*      FDB 616,830,1804,2256,6614
*PRMS  FDB 31 K=9
*      FDB 0,1,2,3,6,7,11,14,17,33,42,43,63,65,67,81,134,162,206,211
*      FDB 366,663,782,1305,1411,1494,2297,2826,3230,3354,3417,3698

SIZE  EQU 13396  must be multiple of 2, see DBL
NUM   RMB SIZE  space for number in decimal, *.3010344497+1
NUMEND EQU *  1+other end of number

* Primes taken from various issues of Mathematics of Computation
* Also see Recreational Mathematics by Beiler
 END START
```

```
Perfect number for n=2, (2**n-1)*2**(n-1) =
6

Perfect number for n=3, (2**n-1)*2**(n-1) =
28

Perfect number for n=5, (2**n-1)*2**(n-1) =
496

Perfect number for n=7, (2**n-1)*2**(n-1) =
8128
```

Perfect number for n=13, (2**n-1)*2**(n-1) =
33550336

Perfect number for n=17, (2**n-1)*2**(n-1) =
8589869056

Perfect number for n=19, (2**n-1)*2**(n-1) =
137438691328

Perfect number for n=31, (2**n-1)*2**(n-1) =
2305843008139952128

Perfect number for n=61, (2**n-1)*2**(n-1) =
2658455991569831744654692615953842176

Perfect number for n=89, (2**n-1)*2**(n-1) =
191561942608236107294793378084303638130997321548169216

Perfect number for n=107, (2**n-1)*2**(n-1) =
13164036458569648337239753460458722910223472318386943
117783728128

Perfect number for n=127, (2**n-1)*2**(n-1) =
14474011154664524427946373126085988481573677491474835
889066354349131199152128

Perfect number for n=521, (2**n-1)*2**(n-1) =
23562723457267347065789548996709904988477547858392600
710143027597506337283178622239730365539602605615602
555646625032701750528925780432155433824984287771524
270103944969186640286445341280358314397902368386240331
714359223566432197031017207135163527487298747400647801
939587165936401087419376496057918549492160555646976

Perfect number for n=607, (2**n-1)*2**(n-1) =
141053783706712069063207958086063189881486743514715667
838838675999954867742652380114104193329037690251561 9
505687098293271640877243663700871167312681593136524874
506524398058772962072974467232951666582288469268077
866528701889208678794514783646593139220603706950647 3
607357237869517647305526682625328488638371507297452 4
463835300053138429460296575143368065570759537328128

Editor's Note: Due to the length of those numbers between this point in the numbers listing, and the final number, they have been deleted. The skips in the final number n=44497 are caused, not by the program, but by our lack of sufficient memory to hold the entire number in one block. The printer reset itself to the beginning of a new line as we loaded in additional portions of the number. Therefore, the number (yep, it is just one long, and I **mean LONG** number) is all that follows below!! So you FORTH and PERFECT NUMBERS types have fun. If you prove this wrong, I might publish something concerning the error, but no more long numbers. One more number and it would have filled up the entire magazine plus!

DMW - - -

DMS NOTES

Bill Adams
21 Larch Road
Briarcliff Manor, N.Y. 10510

This month we will examine an application which is familiar to us all - the checkbook. I have selected this application since it brings out some of the additional capabilities of a good DMS system. The examples used below will be quite familiar to users of DMS2/VM as they are borrowed from the sample application in the user guide. A few variations have been added for the purpose of this article.

Without spending a lot of time on rationale I will simply begin by defining a file which resembles the standard check ledger. To save space, the input session is omitted and the file definition shown below.

CHEX80 SPECIFICATION			INTERNAL		EXTERNAL	
LEV	GRP FLD	FMT	OFS	LEN	OFS	LEN
***	*** ********	***	***	***	***	***
. 1	CHEX80	C,8				
.	MO	N,2	0	1	0	2
.	DA	N,2	1	1	3	2
.	YR	N,2	2	1	6	2
.	CHK0	N,4	3	2	9	4
.	ACCOUNT	A,24	5	24	14	24
.	AMOUNT	D,4.2	29	3	39	9
.	DEPOSIT	D,4.2	32	3	49	9
.	BALANCE	D,4.2	35	3	59	9
.				---		---
TOTAL RECORD LENGTH:				38		68

Incidentally, users with 64 character screens may wish to shorten the ACCOUNT field in order to use the horizontal "tabular" format when updating the file. Users with 32 character screens must use the "vertical" update mode.

Now that we have defined the checking file, we may enter and list our transactions. Note that I have entered a zero CHK# entry for deposits and have included a balance forward entry as the first record.

```
GENER V2.1   DATE: 10-03-82          SOURCE: CHEX80      PAGE: 0001

CHEX80
MO DA YR CHK# ACCOUNT              AMOUNT    DEPOSIT   BALANCE
== == == ==== ------------------- --------- --------- ---------
00 00 00 0000 BALANCE FORWARD         .00    359.62       .00
01 01 80 0000                         .00    550.00       .00
01 01 80 0231 CASH                  150.00       .00       .00
01 01 80 0232 GASOLINE               34.60       .00       .00
01 01 80 0233 WATER                  27.80       .00       .00
01 01 80 0234 TELEPHONE              32.45       .00       .00
01 01 80 0235 HEATING               112.50       .00       .00
01 10 80 0236 VISA                  100.00       .00       .00
01 15 80 0237 CASH                  125.00       .00       .00
02 01 80 0000                         .00    550.00       .00
02 01 80 0238 GASOLINE               44.50       .00       .00
02 01 80 0239 HEATING               112.50       .00       .00
02 01 80 0240 TELEPHONE              26.40       .00       .00
02 01 80 0241 CASH                  150.00       .00       .00
02 10 80 0242 VISA                  100.00       .00       .00
02 10 80 0243 INSURANCE-LIFE         45.65       .00       .00
02 15 80 0244 CASH                  100.00       .00       .00
--(etc)--
```

The BALANCE column is not filled for a reason! The computer is going to figure it out (which is why I included the balance forward entry). This is accomplished by the following .CTL file which may be executed after each posting (or correction) of the file.

```
FILE CHEX80, COPY, NAME CHEX80,
CALC REG1 = DEPOSIT - AMOUNT + REG1
CALC REG1 = BALANCE
END
```

Execution is accomplished by entry of "GENER" and the control file name. Note that the CHEX80 file is overlayed with each execution as is the amount in the BALANCE column. The revised file then looks like the following.

```
GENER V2.1  DATE: 10-03-82          SOURCE: CHEX80      PAGE: 0001

CHEX80
MO DA YR CHK# ACCOUNT                    AMOUNT    DEPOSIT   BALANCE
-- -- -- ---- ------------------------ --------- --------- ---------
00 00 00 0000 BALANCE FORWARD               .00    359.62    359.62
01 01 00 0000                               .00    550.00    909.62
01 01 00 0231 CASH                       150.00       .00    759.62
01 01 00 0232 GASOLINE                    34.60       .00    725.02
01 01 00 0233 WATER                       27.80       .00    697.22
01 01 00 0234 TELEPHONE                   32.45       .00    664.77
01 01 00 0235 HEATING                    112.50       .00    552.27
01 10 00 0236 VISA                       100.00       .00    452.27
01 15 00 0237 CASH                       125.00       .00    327.27
02 01 00 0000                               .00    550.00    877.27
02 01 00 0238 GASOLINE                    44.50       .00    832.77
02 01 00 0239 HEATING                    112.50       .00    720.27
02 01 00 0240 TELEPHONE                   26.40       .00    693.87
02 01 00 0241 CASH                       150.00       .00    543.87
02 10 00 0242 VISA                       100.00       .00    443.87
02 10 00 0243 INSURANCE-LIFE              45.65       .01    398.22
02 15 00 0244 CASH                       100.00       .00    298.22
--(etc)--
```

Once the checking file is established, we can d some useful manipulation using the GENER program For example, I have found it useful to see th monthly expenses in a matrix format. This is easil accomplished with the following control statements

```
FILE CHEX80
PRINT ACCOUNT
PRINT AMOUNT FOR MO = 01 AND 02 AND 03 AND 04 AND 05 AND 06...
SUM BY ACCOUNT EXC IF CHK# > 0000
TOTAL, TAGS 0, END
```

For purposes here, the matrix is limited to si months by the column width. Users with a 132 columr printer will find that all 12 months will just fi with the field sizes given (24 columns for ACCOUN plus 12 x 9 for AMOUNT = 132). The resulting output appears as follows.

```
GENER V2.1  DATE: 10-03-82          SOURCE: CHEX80      PAGE: 0001

GENER          AMOUNT
ACCOUNT        01       02       03       04       05       06
-------------- -------- -------- -------- -------- -------- --------
CASH           275.00   250.00   295.00   300.00   500.00   250.00
DR. HYDE         .00      .00     55.00      .00      .00      .00
DR. JEEKYL       .00      .00      .00     35.00      .00      .00
GASOLINE       34.60    44.50    39.80    42.50   135.80    29.45
HEATING       112.50   112.50   112.50   112.50   112.50   112.50
INSURANCE-CAR    .00      .00   112.50      .00      .00   112.50
INSURANCE-LIFE   .00    45.65      .00      .00    45.65      .00
TELEPHONE      32.45    26.40    39.80    27.85    23.44    22.90
VISA          100.00   100.00    89.44      .00    37.31      .00
WATER          27.80      .00      .00    27.80      .00      .00
.
.             582.35   579.05   744.04   545.65   854.70   527.35
```

We could get "fancy" and add a title and perhaps rename the columns as "JAN","FEB"... with a second execution. Another possibility is to define a "class" code into our checking file for "MED","AUTO" or other indicators. This would allow summary of these catagories into a matrix and would be useful at tax time. Note that entry of the "SUM BY" fields must be consistant in either case. If space permits, line totals can be also generated using CALC statements. Many other possibilities exist and I will leave them to the imagination of the reader.

With the mailing list discussed last month, we now have two applications for our DMS system. By now the reader should begin to appreciate the flexibility and ease of use that a DMS can offer. If we employed DMS2/VM for both applications, the cost is only $50. each and we have only scratched the surface as to the applicability. In future months, I hope to examine some more applications, thereby showing how the cost per application can be further reduced.

COLOR GRAPHICS

HI - RES ANIMATED
COLOR GRAPHICS

PART II - The ARCADE-50 from TERMINUS DESIGN

Thomas H. Hunt
30001 Wagner
Warren, MI. 48093

A few months ago, a gaping hole in the SS-50's repertoire of boards was filled when TERMINUS DESIGN introduced the ARCADE-50. PART I of this article discussed a major component of the ARCADE-50 -- the TMS9918A Video Display Generator. But that was only part of the story. The complete ARCADE-50 can do much more. It was designed to generate animated color graphics, produce a wide range of music and sound effects, and to provide a sensible human interface -- and it succeeds amazingly well!

The obvious application for this board is, of course, the generation of real-time arcade games, complete with joysticks, fire buttons, and sound effects. However, do not fall into the trap of limiting your imagination to this one specialized area. The addition of graphics to almost any computer application will inevitably enhance its clarity and ease of use. The sound generating capabilities should not be casually dismissed either. Those interested in music composition or synthesis should be able to do wonders with the nine voices and many tonal effects that are available. In fact, your imagination will be your only limitation.

THE HARDWARE

The board itself takes up one SS-50 slot and comes fully assembled and tested. Although the added frills of solder mask and silk screening were foregone, the board, layout and assembly show a better than average level of professionalism. Gold connectors are an option which I decided to omit.

A TMS-9918A chip provides the video capabilities, and three AY-3-8910 Sound Generator chips weave the aural effects. In addition, the board supports several necessary pieces of hardware. Inputs for up to four X-Y proportional joysticks (with pushbuttons) plus eight bits of uncommitted parallel I/O are provided. An audio output hooks directly to an 8 ohm speaker and a composite video output will directly connect to a color monitor -- or, optionally, go to an RF modulator for input into a color TV set. An external video input (from a video camera or VTR) is also available for those who desire to work with this option. Conveniently, an on-board sync separator circuit has also been provided. Access to these inputs and outputs is via three DIP headers.

All the necessary external hardware can be obtained directly from TERMINUS DESIGN at added, but reasonable, cost -- or the user may provide them himself. External cables and connectors come unassembled from TERMINUS DESIGN to allow custom fit for the individual enclosure.

I received the cable and connector set, two Radio Shack joysticks, and a SUP-R-MOD RF modulator. Although the RF modulator is off- board and somewhat inconvenient to mount, I am sold on this arrangement. I have three other color video boards with various RF modulator arrangements, but I consistently get the best picture with the SUP-R-MOD. A little experimenting directly attributed this to three factors:

1). Keeping the modulator and digital circuits physically separate,
2). Effective shielding, and
3). Using UHF channel 33 instead of VHF channel 3.

It took a couple of evenings work to drill some holes, mount and hook up the connectors, and find a convenient place to set the color TV. The board itself offers two jumper options: The vertical retrace signal can go to either FIRQ or IRQ, and the audio output can be either stereo or monaural (there are two on-board one watt audio amplifiers).

A DIP switch allows addressing the board to any 256 byte block. This is somewhat distressing, as it actually uses only six bytes of memory space. Sadly, I have found that all manufacturers tend to cut corners, to some degree, in this area. With today's overworked memory map, this practice must be corrected in any future designs.

THE MANUAL

Most budding manufacturers soon find out, much to their chagrin, that writing an effective instruction manual is no trivial task. Most initial efforts leave a bit to be desired, and the ARCADE-50 manual did not prove to be the exception. Actually, every needed piece of information is there, but one cardinal rule was violated. It is not totally and unequivocally obvious, on first reading, just exactly where to start and what to do with all the information given.

After a couple of readings (the manual is well printed and in very readable type size) things do start falling into place. The first few pages give system requirements, how to hook up the connectors, and initial checkout. Next follows a series of assembly language drivers to handle the various primitive functions of the devices. It will take some study to fully understand their proper use.

Also included is the TMS-9918A VDP programmer's manual and the AY-3-8910 Programmable Sound Generator manual. This encompasses about 60 pages of extremely useful information. It turns out that these have been my primary reference sources while writing programs for the ARCADE-50.

BRINGING IT ON-LINE

Before plugging the board in, a change or two may be in order. First, I found that C11, the video output capacitor, is not only unnecessary, but actually detrimental to video reception. It should be removed and a jumper put in its place. Also, VMA was not connected on some earlier boards. This evidently only poses problems on 6800 systems. TERMINUS DESIGN has now corrected this situation.

After making all necessary connections and plugging everything in, you are now ready for the smoke test. If you are running a standard FLEX09 system (whatever that means) your problems are almost over. A FLEX09 demo disk is included that has several programs which exercise and check out various board functions. Unfortunately, most of these fall into the class of "Not-Ready-For-Prime-Time" programs. Instructions for use are too casually mentioned in the manual and the programs themselves are not really production-ready. Some of them prompt for a board address, others require editing. Some have ill-defined and unexpected inputs and outputs. I finally ended up listing them all, deciding what they were supposed to do, making changes, and then running them. Terminus Design now recognizes these problems and is in the process of preparing a more meaningful demo disk.

There is one program on the demo disk that does deserve praiseworthy mention. It is a very good PAC-MAN type game, complete with sound effects, called RATMAZE. It operates under Terminus Design's proprietary real-time operating system (object included) and requires IRQ and a serial terminal. The only problem encountered was that the program expects the console ACIA to be addressed at $E004. My system is (to put it mildly!) not exactly standard anything. I had to change seven absolute addresses in RATMAZE that referenced the ACIA. For those with a similar problem, the seven addresses are:

$0839 $0970 $0982
$093E $097F $0958
$0941

Once the RATMAZE program is running, it is singularly impressive. The smooth animation, sound, and color only served to whet my appetite to create bigger and better programs. I was further impressed by the fact that all this was now happening on MY computer! Even my children have started begging to learn about the computer (sometimes I wonder if this is really a step forward). Now that everything is running and the capabilities of the ARCADE-50 actually sink in, all problems have dwindled into perspective -- minor inconveniences only!

SOUND GENERATING CAPABILITIES

It is sometimes difficult to decide whether the ARCADE-50 is primarily a graphics board or primarily a sound generating board. This paradox is created by the presence of three General Instruments AY-3-8910 Programmable Sound Generator (PSG) chips. With two LM386 audio amplifiers on-board, it is only necessary to connect the appropriate speakers to obtain musical or special sound effects.

The ARCADE-50 offers a stereo or monaural option, selected by jumpers. In monaural, the outputs of all three PSG's are mixed into one amplifier. For stereo, two PSG's are mixed into one amplifier, while the other amplifier is dedicated to the third PSG. Stereo, of course, will require the addition of two speakers.

For those unfamiliar with the AY-3-8910, it is a register oriented sound generator. That is, all control commands are given to the PSG by writing to one of its sixteen address registers. Once the registers are loaded, the PSG takes over to generate and sustain the programmed sound. The system processor is then free to perform other tasks.

Each PSG contains three separate tone generators which have a range of subaudio to about 75,000 cycles. Each channel also has its own amplitude register, programmable to one of sixteen levels. I am estimating a dynamic range of the total circuit to be only about 35 db. -- not the best, but certainly far from being unacceptable.

There are two other special effects items on each PSG that have a global effect on all three tone channels. One is a pseudo-random noise generator that can be mixed in varying amounts with the tone generators. This is very useful in creating effects

such as explosions, gun shots, snare drums, and steam locomotives. The other necessary tool available is the Envelope Generator. This allows the programmer to have control over the attack, sustain, and decay of the particular sound.

Add three such PSG's onto one board and you have some very exciting potential. For those interested in musical composition or synthesis, the nine separate tone channels offer a wide range of possibilities. I have heard some rather impressive music generated on the ATARI 800, which has only four tone channels. I can only imagine what some creative individuals will do with nine channels available!

SOFTWARE AVAILABLE

Inevitably, the more complex and flexible a board is, the more heavily it relies on software. It would be absurd to expect a small manufacturer to have a full catalogue of applications programs instantly available. Instead Terminus Design chose, and correctly so, to concentrate their efforts towards providing the user with adequate tools to generate quality applications programs.

The tried-and-true route is to provide a high level language -- usually BASIC -- with a set of built in commands for graphics and sound manipulations. Terminus Design went one step further and secured a BASIC Compiler! FBASIC, as Terminus Design calls it, is in reality an enhancement of Microware's A-BASIC compiler. About 31 commands were added to A-BASIC for controlling the graphics and sound functions of the ARCADE-50. FBASIC also retains all of the original A-BASIC functions, as well as allowing disk I/O through FLEX.

While FBASIC is a minimal implementation of Basic, with only 16 bit integer math and no scientific functions, it makes ultimate sense for an application like this. The primary advantage is program execution speed. A Basic interpreter is almost invariably too slow to produce a meaningful real-time graphics display. Also, arcade games seldom compute any complex functions "on the fly". Thus, the FBASIC math package is more than adequate for most applications. Finally, as it is a high level language, program development time is drastically reduced.

It is totally beyond the scope of this article to discuss all the nuiances of FBASIC -- I will reserve that for future topics. I will say that, even though it has a few problems, I have found it to be an invaluable development tool. I strongly recommend this option to anyone purchasing the ARCADE-50.

Presently, the only alternative to FBASIC is to use some assembly language drivers. A minimal set of drivers is included with the ARCADE-50 documentation. I have written a more extensive driver set that handles all functions in all modes and occupies a little less than 2K of memory. These could be used with assembly language applications programs or patched into an interpreted Basic via USER calls. The source code is in the hands of Terminus Design and may be available through them.

Terminus Design is also setting up a user's group for applications software. I have seen several of the initial efforts and they all look promising. Included is the already-mentioned RATMAZE game, an OTHELLO game, a SPRITE CREATOR/EDITOR program, and a program that allows drawing figures with the joystick and creating three dimensional images. Consult Terminus Design for availability.

CONCLUSION

The difficult part about writing a review of a board like the ARCADE-50 is answering the question -- "Where does one stop?". With boards of this potential and versatility there will always be just one more function to describe or just one more point to clear up. Perhaps that is what is so interesting about graphics and sound -- the continual challenge. After all, what can be said about a memory board, an I/O port, or a disk controller? Once operational, they cease to be entities in themselves, and simply blend invisibly into the system.

Ah, but boards like the ARCADE-50 are an entirely different story! They continually make their presence known, constantly begging for bigger and better programs. They will continually test your ability and ingenuity as a programmer and, hopefully, inspire you to rise to the occasion.

UNIFLEX FORTRAN UTILITIES

STRING-HANDLING UTILITIES FOR UNIFLEX FORTRAN

by Art Mathews
University of South Florida
Biology Dept., LIF 169
Tampa, FL 33620

For computer-assisted instruction, the application on which I work, you need to be able to easily deal with character strings. Of all the languages you can pick from, FORTRAN is probably the most unlikely choice for such an application, but for several years, we were time-sharing on a computer on which FORTRAN was by far the most flexible language. That computer had a large library containing just about every utility you could ever want. Over the years, I wrote dozens and dozens of FORTRAN programs, all using these utility subs. When we got our own Southwest Tech 8/09 computer and UniFLEX FORTRAN, it was clear that the first thing I had to do was to write utilities just like the ones on the old computer. Once that was done, the conversion of the application programs was a fairly straightforward chore.

This paper describes 13 of those utility subs which I think would be useful to anyone working with this version of FORTRAN. Most of the subs are written in source code for the relocating assembler, and rest are in FORTRAN. The general procedure for incorporating machine-language utilities into UniFLEX FORTRAN was described in an earlier paper of mine published in this journal.

UniFLEX is, of course, a multi-user operating system of Technical Systems Consultants, and a trademark of theirs. TSC released FORTRAN 77 for UniFLEX early in 1982. To use FORTRAN 77 you also need the relocating assembler and the linking loader.

In FORTRAN 77 one can declare a character string of any size. For example, the declaration:

```
      character*40 LINE
```

defines a string called 'LINE' which contains a maximum of 40 characters. I am going to call this type of variable a character 'string' in contrast to a character 'array'. An example of a character array declaration is the following:

```
      character*1 TEXT(40)
```

Most of the routines below deal with strings rather than arrays, but two conversion routines are included: STOA converts a string into an array, and ATOS converts an array into a string.

User input is usually done as follows:

```
      character*40 LINE
         . . .
      read(5,100) LINE
  100 format(a40)
```

The read statement will accept just about anything the user types. In CAI you must not force the user to format his or her input in any particular way! Once the program has the input string, it can procede to analyze it to see if it makes sense in the current situation. For example, if the user was supposed to input an integer value, the program can use the NUMGET function to extract that value.

The string 'LINE' in the read statement above will receive the first 40 characters that the user types. If the user types less than 40 characters, the string is padded on

the right with spaces. NO END OF STRING TOKEN IS INSERTED! Therefore we will deal with strings in this format. No end of string token will be used, and strings will be padded on the right with spaces.

The source files are given in the two listings. The filename of Listing 1 is 'package1'. This is a source file for the relocating assembler. The Fortran source code in Listing 2 should be named 'package2.f'. The following commands will create a library called 'package' which contains all of the subprograms:

```
relassb package1
f77 package2.f -c
lib-gen n=/lib/package u=package1.r u=package2.r
```

Once this library is created in this manner, it can be accessed by any user via the 'link-edit' command as described in Chapter 2 of the manual entitled 'UniFLEX Linkage Editor'.

What follows is a description of each of the routines. Copies of this may be distributed to all of your FORTRAN users.

```
**************************
String-Handling Utilities
**************************
```

For an explanation of the use of libraries, read Chapter 2 of 'UniFLEX Linkage Editor.'

This information sheet describes the routines in the library, '/lib/libxx'. These routines deal mainly with character strings. Since strings can be used as the filename in an OPEN statement, these routines help to manipulate filenames, but their presence also makes it practical to use strings instead of character arrays.

All Fortran programs require the library, '/lib/F77.runlib'.

Definitions:

A character STRING is a character variable of size greater than one. For example, the following declaration forms a character string of size 40:

```
character*40 LINE
```

The SIZE of a string is the size specified in its declaration.

The LENGTH of a string is the number of characters up to the last non-blank character. Note that there is no end-of-string marker in filenames or any other strings.

A character ARRAY is an array of characters or character strings. Note the difference in the following array declaration from the previous string declaration:

```
character*1 TEXT(40)
```

Both LINE and TEXT contain exactly 40 characters, but they are stored differently in memory.

The POSITION of a character within a string is 1 for the first character.

1. subroutine CFILL(KHAR,STR,IPOS,NCH)

   Fills string STR with character KHAR.
   IPOS = beginning position within STR where first character will be placed.
   NCH = number of characters to be replaced.

   For example, to fill an entire string with blanks:

```
character*50 line
call cfill(' ',line,1,50)
```

2. subroutine CMOVE(STR1,IPOS1,STR2,IPOS2,NCH)

   Copies characters from STR1 to STR2.
   STR1 = character string to be copied FROM.
   IPOS1 = position within STR1 of the first character to be copied.
   STR2 = character string to receive the characters.
   IPOS2 = position within STR2 where first character will be placed.
   NCH = number of characters to be copied.

3. integer function NACS(LINE,LSIZE)

   Find the length of a string. Actually, what it finds is the last non-blank character in the string.
   LINE = character string.
   LSIZE = size of the string.

   For example, to find the number of characters typed by user:

```
character*40 line
read(5,1) line
1 format(a40)
len=nacs(line,40)
```

   Note: LSIZE must equal the declared string size.

4. subroutine CONCAT(STR1,L1,STR2,L2,STR3,L3)

   Sets STR3 = concatenation of STR1 and STR2.
   L1 = declared size of STR1.
   L2 = declared size of STR2.
   L3 = declared size of STR3.

   1. finds length of STR1 (using NACS)
   2. moves that many characters to STR3
   3. finds length of STR2
   4. moves that many characters to STR3
   5. fills the rest of STR3 with blanks

   If the length of STR1 + length of STR2 is greater than the size of STR3, then appropriate truncation will occur.

5. subroutine LPARSE(LINE,LSIZE,KPTR,IBEG,JSIZE)

   Parses fields within a string. Can be used iteratively to parse all the fields in the string. Fields are separated by blanks or a single comma. Extra blanks are ignored.
   LINE = character string.
   LSIZE = declared size of LINE.
   KPTR = sliding pointer. This must be a VARIABLE, never a constant! It should be initialized to the first character to examine. The routine increments this variable as it processes the string and returns with KPTR pointing to the first character after the last character in the field.
   IBEG = integer VARIABLE! Returns as the position of the first character of the next field, or as zero if there were no more fields in the string.
   JSIZE = integer VARIABLE! Returns as the number of characters in the field which starts at IBEG, or as zero if the field was null (i.e. ',,') or not found.

6. integer function MATCH(LINE,LSIZE,STR,NCH)

   Determines whether STR is a substring of LINE.
   LINE = the longer character string.
   LSIZE = declared size of LINE.
   STR = the shorter character string.
   NCH = number of characters of STR to check.

   MATCH will return with a value of zero if STR does not match any part of LINE. If it does match, the value is the position of the first character of LINE that matches the first character of STR.

7. subroutine LEFT(STR1,STR2,LSIZE,NCH)

   Sets STR2 = the left NCH characters of STR1
   LSIZE = declared size of STR2

   The NCH left-most characters of STR1 are transferred to STR2 and the remainder of STR2 is filled with blanks. If NCH is greater than LSIZE, then STR2 equals the left-most LSIZE characters of STR1. If NCH = 0, then STR2 will be filled with blanks.

8. subroutine MID(STR1,IPOS1,LSIZE1,STR2,LSIZE2,NCH)

   Sets STR2 = a portion of STR1.
   IPOS1 = position within STR1 of where the new string starts.
   LSIZE1 = declared size of STR1.
   LSIZE2 = declared size of STR2.
   STR2 includes NCH characters from STR1. The remainder of STR2 is filled with blanks.

9. integer function ICOMP(STR1,IPOS1,STR2,IPOS2,NCH)

   Compares two strings.
   IPOS1 = position within STR1 of first character of the field.
   IPOS2 = position within STR2 of first character of the field.
   NCH is the number of characters to compare.

   If the STR1 field is identical to the STR2 field then ICOMP = 0.
   If the STR1 field precedes the STR2 field in alphabetical order, then ICOMP = 1.
   If the STR1 field follows the STR2 field in alphabetical order, then ICOMP = -1.

10. integer function NUMGET(LINE,IPTR,NCH)

    Extracts a decimal integer from a string.
    LINE = character string containing an integer represented by decimal digits.
    IPTR = sliding pointer. This must be a VARIABLE, never a constant! It should be initialized to the position within LINE of the first character to examine. It will be incremented to one place beyond the last digit of the number.
    NCH = Maximum number of characters to examine before stopping.
    NUMGET is assigned the value of the integer found or zero if no digits were found.

    Leading spaces or commas are ignored. The scan terminates on any of the following conditions: (1) NCH characters have been examined, (2) a character is encountered which not a digit, space, or comma, or (3) a space or comma is encountered after a group of digits.

11. subroutine BTOA(LINE,ARRAY,LSIZE)

    Converts a character string to a character array.
    LINE = character string.
    ARRAY = character array.
    LSIZE = number of characters to convert.

    For example:

```
character*50 LINE
character*1 ARRAY(50)
.
.
.
call btoa(LINE,ARRAY,50)
```

```
      Both LINE and ARRAY contain exactly 50 characters, but
      they are stored differently in memory.

  12. subroutine ATOS(LINE,ARRAY,LSIZE)

      Converts a character array to a character string.
      LINE = character string.
      ARRAY = character array.
      LSIZE = number of characters to convert.

      This is the opposite of STOA.

  13. subroutine CHR(CHAR,INT)

      Lets CHAR = the character whose ASCII value is INT.

      This is the opposite of the intrinsic function, ICHAR.

* LISTING 1: Source file for relocating assembler

 name _package1

* * *  CFILL  * * *
*
* subroutine cfill(KHAR,STR,IPOS,NCH)

 global _cfill
 text
_cfill leax 2,s x points to argument list
 pshs d,u

* Find STR position

 ldd 2,x STR base address
 addd [4,x] IPOS
 subd #1
 tfr d,u u points to STR character

* Fill the string

 lda [,x] fill character, KHAR
 ldx [6,x] x is the character counter
 cmpx #0 NCH must be positive
 ble cfill2 if not, then leave STR alone
cfill1 sta ,u+
 leax -1,x decrement the counter
 bne cfill1 next character

* Clean up the stack & return

cfill2 puls d,u,pc

* * *  CMOVE  * * *
*
* subroutine CMOVE(STR1,IPOS1,STR2,IPOS2,NCH)

 global _cmove
 text
_cmove leax 2,s x points to argument list
 pshs d,x,u

* Find STR2 position

 ldd 4,x STR2 address
 addd [6,x] IPOS2
 subd #1
 tfr d,u u points to STR2 character

* Find STR1 position

 ldd ,x STR1 address
 addd [2,x] IPOS1
 subd #1
 tfr d,x x points to STR1 character

* Move characters

 ldx [8,x] x is the byte counter
cmove1 leax -1,x decrement counter
 cmpx #0 set N flag
 bmi cmove2 done yet?
 lda ,u+
 sta ,x+
 bra cmove1 next character

* Clean up the stack & return

cmove2 puls d,x,u,pc

* * *  NACS  * * *
*
* Number of Actual Characters in String
*
* integer function NACS(LINE,LSIZE)

 global _nacs
 text
_nacs leax 2,s x points to argument list
 pshs d,x,u save registers

* Set up pointers and counters

 ldd ,x LINE address
 addd [2,x] LSIZE
 tfr d,u u points to LINE character
 ldv [2,x] v is the counter

* Scan LINE in reverse order
* Look for first non-blank character

nacs1 lda ,-u
 cmpa #$20 blank
 bne nacs2
 leav -1,v decrement counter
 bne nacs1

* Leave result in the stack & return

nacs2 stv 4,x
 puls d,x,u,pc

* * *  LPARSE  * * *
*
* subroutine LPARSE(LINE,LSIZE,KPTR,IBEG,JSIZE)
*
 global _lparse
 text
_lparse leax 2,s x points to argument list
 pshs d,x,y,u

* Initialize IBEG and JSIZE

 ldd #0
 std [6,x] IBEG
 std [8,x] JSIZE

* Find LINE position

 ldd ,x LINE address
 addd [4,x] KPTR
 subd #1
 tfr d,u u points to LINE character

* Find number of characters remaining

 ldd [2,x] LSIZE, total number of chars
 addd #1
 subd [4,x] KPTR
 bls parse6 make sure it is positive!
 tfr d,v v is the number of chars left

* Look for next non-blank character

parse1 lda ,u+ get next char
 cmpa #$20 blank
 beq parse2
 cmpa #$2c comma
 bne parse3

* The character is a blank or a comma

parse2 leav -1,v decrement char counter
 bne parse1 end of LINE?

* End of Line, KPTR=LSIZE+1

parse7 ldd [2,x]
 addd #1
 std [4,x]
 bra parse6

* Save IBEG value

parse3 tfr u,d
 subd ,x
 std [6,x] IBEG

* Count chars until next space or comma

parse4 ldd [8,x] increment JSIZE
 addd #1
 std [8,x]
 leav -1,v decrement char counter
 beq parse7 end of line?
 lda ,u+ next character
 cmpa #$20 space
 beq parse3
 cmpa #$2c comma
 bne parse4

* End of field, save KPTR

parse5 tfr u,d
 subd ,x
 std [4,x] KPTR

* Clean up the stack & return

parse6 puls d,x,u,pc

* * *  MATCH  * * *
*
* INTEGER FUNCTION MATCH(LINE,LSIZE,STR,NCH)

 global _match
 text
_match leax 2,s x points to argument list
 pshs d,x,u

* Initialize MATCH

 ldd #0
 std B,x MATCH

* MATCH + NCH > LSIZE ?

match1 ldd B,x MATCH
 addd [6,x] NCH
 cmpd [2,x] LSIZE
 bhi match3 if so, abort

* No, keep going
```

THE COMPLETE BUSINESS SYSTEM

+Multiuser+Highly Expandable+Cost Effective

S+ THE CONCEPT

The S+ system is a modular computer system in which all portions of the hardware and software are designed to work together in the most efficient way possible. An S+ single user system with floppy disk storage is a competitive and cost effective entry level system. Unlike most other small computers being sold as "personal", or "small business" machines, the S+ system may be expanded to maximum capabilities using this same hardware and software. You cannot end up with a DEAD END system that cannot be expanded and whose software is not compatible with larger machines. A basic S+ system may be expanded to thirty-two users, a megabyte of main memory and hundreds of megabytes of hard disk storage by simply plugging in, or connecting the desired upgrade equipment.

TOTAL DESIGN—Hardware and Software

The S+ system is an integrated hardware and software design. The two complement and enhance each other in this system. The UniFLEX® operating system used in the S+ systems is patterned after the Bell Laboratories UNIX® operating system, one of the most admired and widely used operating systems in the world. Instead of being an afterthought, the software is part of the design of the S+ system. You can be sure that with this approach that all parts of the computer operate with maximum efficiency and cost effectiveness.

THE CENTRAL PROCESSOR

The basic S+ system is configured with 256K bytes of memory and can be expanded to more than 1 million bytes. An efficient and fast hardware memory management system is used to allocate the available memory among the users on a dynamic basis. As little as 8K bytes, or the entire memory—if needed—can be used by any individual user. This makes it possible to run very large programs on the system, but it also uses no more memory than necessary for a particular job. The increase in cost effectiveness of this system over crude and outdated bank switching arrangements is dramatic.

The central processor runs in both user and supervisor states. It can detect and reject a defective user program. It is impossible for a user program to go bad and stop the entire system, as can happen quite easily in less sophisticated systems.

Task switching is accomplished by use of a multiple map RAM memory, with sixty-four individual task maps. Each task can access from 4 to 64 K-bytes of memory. Multiple tasks may be used in programs that require more than 64K bytes of memory for execution. When a task is completed the memory is automatically released for other use.

SOFTWARE

The S+ operating system, UniFLEX® is a multiuser, multitasking operating system based on the UNIX® operating system that has been used for many years on Digital Equipment Corp. PDP-11 series minicomputers. It is considered one of the most sophisticated and "user friendly" operating systems available. Variations of UNIX® are rapidly becoming standard on mini and larger microcomputers.

A large variety of languages are available for use with the system. These include FORTRAN, COBOL, BASIC, and Pascal. Word processing packages are also available to give you full text processing capability on the system.

Applications programs are available in large quantities in many fields. This includes general business, medical, dental, veterinary, library and real estate management; plus others. Since the system is multiuser it can also be connected to cash registers to produce a point-ofsale terminal system combined with the computer. The possibilities for application of this system are endless.

THE I/O SYSTEM

The S+ system is totally interrupt driven. All terminal and printer I/O devices connect to an I/O bus separate from the main bus. Up to thirty-two separate devices may be connected to the I/O bus at any one time. If I/O activity is great enough to cause an unacceptable slowdown in system operation, a separate I/O processor can be installed in the system. This plug-in option removes all I/O handling overhead from the main processor and allows operation of up to thirty-two external devices at 9,600 baud. Without an integrated total design, as in the S+ system, it would become impractical to use a UNIX®type operating system in a situation with heavy terminal I/O activity.

DISK STORAGE

A wide range of disk storage capacity is available for the S+ system, from 2.5 M-byte floppy disks to an 80 M-byte Winchester and many sizes between. All disk controllers use direct memory access (DMA) type operations to maximize data transfer and to minimize overhead on the main processor. The Winchester disks also use intelligent controllers along with DMA transfers to preserve the performance that these type devices are capable of giving. Without this distributed intelligence the system performance would be greatly degraded. The UniFLEX®operating system is designed to work at maximum efficiency with this type disk system. The data transfer rates achieved by this combination rival those of large minicomputers.

COMMUNICATIONS

A high speed local network communications system is available to interconnect S+ systems. The VIABUS® network will allow communication between systems at data rates of over 400K baud. Such a system makes it possible to share data between local systems in an efficient and low-cost manner.

AVAILABLE SOON

Tape backup—20M-Byte in less than 15 minutes on a standard ¼ inch cartridge.

Mini-Wini—5 and 10 M-Byte Winchesters—5¼ inch package. Winchester performance, for smaller systems in a small package. UniFLEX® compatible design.

Large Capacity—190 and 340 M-Byte Winchesters, plus SMD cartridge drives.

UniFLEX is a registered trademark of Technical Systems Consultants, Inc.

UNIX is a registered trademark of Bell Labs.

VIABUS is a registered trademark of Southwest Technical Products Corporation.

```
 ldd 8,x MATCH
 addd ,x LINE address
 tfr d,y y points to LINE char
 ldd 8,x increment MATCH
 addd #1
 std 8,x
 ldu 4,x u points to STR char
 ldd [6,x] NCH
 cmpa #0 >255 ?
 bne match3 if so, abort

* Compare LINE and STR

match2 lda ,u+ STR char
 cmpa ,y+ same as LINE char?
 bne match1 if not, shift LINE
 decb end of STR?
 bne match2 if not, keep going
 bra match4 yes, return

* No match, MATCH=0

match3 ldd #0
 std 8,x MATCH

* Clean up the stack & return

match4 puls d,x,y,u,pc

* * *  ICOMP  * * *
*
* integer function ICOMP(STR1,IPOS1,STR2,IPOS2,NCH)

 global _icomp
 text
_icomp leax 2,s x points to argument list
 pshs d,x,y,u

* Find STR2 position

 ldd 4,x STR2 address
 addd [6,x] IPOS2
 subd #1
 tfr d,u u-reg points to STR2 character

* Find STR1 position

 ldd ,x STR1 address
 addd [2,x] IPOS1
 subd #1
 tfr d,y y-reg points to STR1 character

* Compare characters

 ldx [8,x] x is now the character counter
icomp1 leax -1,x decrement counter
 cmpx #0 set N flag
 bmi icomp3 identical strings?
 lda ,y+
 cmpa ,u+
 beq icomp1 next character

* Characters are different

 bhi icomp2 are they in order?
 ldd #1 yes
 bra icomp4
icomp2 ldd #$ffff minus 1
 bra icomp4

* Identical strings

icomp3 ldd #0

* Leave result in the stack & return

icomp4 std 10,s
 puls d,x,y,u,pc
* * *  STOA  * * *
*
* subroutine STOA(LINE,ARRAY,LSIZE)

 global _stoa
 text
_stoa leax 2,s x points to argument list
 pshs x,y,u

* Set pointers

 ldy ,x LINE address
 ldu [2,x] ARRAY address

* Move characters

 ldx [4,x] x is now the byte counter
stoa1 leax -1,x decrement counter
 cmpx #0 set N flag
 bmi stoa2 done yet?
 lda ,y+
 sta ,u+
 bra stoa1 next character

* Clean up the stack & return

stoa2 puls x,y,u,pc

* * *  ATOB  * * *
*
* subroutine ATOB(LINE,ARRAY,LSIZE)

 global _atob
 text
_atob leax 2,s x points to argument list
 pshs x,y,u

* Set pointers

 ldy ,x LINE address
 ldu [2,x] ARRAY address

* Move characters

 ldx [4,x] x is now the byte counter
atob1 leax -1,x decrement counter
 cmpx #0 set N flag
 bmi atob2 done yet?
 lda ,y+
 sta ,u+
 bra atob1 next character

* Clean up the stack & return

atob2 puls x,y,u,pc

* * *  CHR  * * *
*
* subroutine CHR(CHAR,INT)

 global _chr
 text
_chr leax 2,s x points to the argument list
 pshs d
 ldd [2,x] get ascii value INT
 stb [,x] save CHAR
 puls d,pc return

 end
++

* LISTING 2: FORTRAN source code
*
* * *  CONCAT  * * *
*
* Concatenate str1 and str2 to form str3
*
      subroutine concat(str1,l1,str2,l2,str3,l3)
      external nocs,cmove,cfill
      integer nocs
      character*1 str1,str2,str3
      len1=nocs(str1,l1)
      len2=nocs(str2,l2)
      if(len1.gt.l3) len1=l3
      call cmove(str1,1,str3,1,len1)
      ipos3=len1+1
      if(len1+len2.gt.l3) len2=l3-len1
      call cmove(str2,1,str3,ipos3,len2)
      nch=l3-len1-len2
      ipos3=len1+len2+1
      call cfill(' ',str3,ipos3,nch)
      return
      end
* * *  LEFT  * * *
*
* Let STR2 = leftmost NCH characters of STR1
*
      subroutine left(str1,str2,lsize,nch)
      external cmove,cfill
      character*1 str1,str2
      n=nch
      if(n.gt.lsize) n=lsize
      call cmove(str1,1,str2,1,n)
      l=lsize-n
      n=n+1
      call cfill(' ',str2,n,l)
      return
      end
* * *  MID  * * *
*
* Move NCH characters from STR1 to STR2 beginning at
* position IPOS1 of STR1.
*
      subroutine mid(str1,ipos1,lsize1,str2,lsize2,nch)
      n=nch
      if(ipos1+n.gt.lsize1) n=lsize1+1-ipos1
      if(n.gt.lsize2) n=lsize2
      call cmove(str1,ipos1,str2,1,n)
      l=lsize2-n
      n=n+1
      call cfill(' ',str2,n,l)
      return
      end
* * *  NUMGET  * * *
*
* Extract a decimal integer from LINE
*
      integer function numget(line,iptr,nch)
      character*1 line,kh
      external cmove
      j=0
      k=nch
      numget=0
    1 k=k-1
      if(k.lt.0) go to 2
      call cmove(line,iptr,kh,1,1)
      iptr=iptr+1
      if(j.eq.0 .and. (kh.eq.' ' .or. kh.eq.',')) go to 1
      j=1
      i=ichar(kh)-48
      if(i.lt.0 .or. i.gt.9) go to 2
      numget=10*numget+i
      go to 1
    2 return
      end
++
```

INTRODUCING THE MC68000 EDUCATIONAL COMPUTER

BY

Mary Gallagher
MOTOROLA, INCORPORATED
SEMICONDUCTOR PRODUCT SECTOR

As the MC68000 microprocessor has become more and more of an industry standard for 16-bit processors, there has been a corresponding growth in customer requests for low cost educational tools based on the MC68000. The MC68000 Educational Computer Board (MEX68KECB) is Motorola's response to these requests. The MEX68KECB is a single-board MC68000-based computer designed to acquaint professionals and advanced university students with the intricacies of MC68000-based systems. The Educational Computer aids the professional in becoming familiar with the characteristics and capabilities of the powerful MC68000, from its instruction set and programming to system integration.

A significant factor in the customer requests was a requirement for low cost. With this in mind, the Educational Computer was designed to sell for no more than $500.

In addition to the board, a dumb RS232C-compatible terminal and a power supply (5V @ 1A, ± 12V @ 50mA) are required.

The Educational Computer Board's hardware/firmware complement offers a lot of capability and a host of functional options. At its heart is a 4MHz version of the MC68000 MPU. Supporting the MPU is a block of two 64K ROMs/EPROMs that provide 16K bytes of firmware. Appropriately called TUTOR, this firmware includes many capabilities designed specifically for educational purposes - a monitor, a debug tool, and an assembler/disassembler.

The monitor/debug portion of TUTOR evolved from MACSbug™ and VERSAbug™, the monitors developed for the M68000 EXORmacs™ and VERSAmodule™ systems. Hence, TUTOR supports all of the MACSbug/VERSAbug commands except those deliberately eliminated because of the hardware limitations of the board (disk I/O for example). But, the command format and syntax are identical to MACSbug/VERSAbug to permit easy upgrading to these systems, if desired later.

The assembler/disassembler portion of TUTOR is not based on either MACSbug or VERSAbug; it is a truly unique feature of TUTOR which allows a user to begin writing programs immediately in assembly language, avoiding the frustration of hand assembly.

This assembler uses the same command line format and syntax as the MC68000 resident and cross assemblers but differs from these other assemblers in several ways. This assembler is a one-line assembler; that is, each source line is immediately assembled into object code and saved on a line by line basis as it is entered. The source lines are not saved. In order to display a program, the object code must first be translated back into source lines using the disassembler portion of TUTOR.

The MEX68KECB assembler recognizes all MC68000 instruction and addressing mnemonics; labels, line numbers, and comments are not allowed. One assembler directive is also included to provide a way of entering data.

In addition to the on-board firmware, the Educational Board contains 32K of dynamic RAM. The array is organized as a 16K x 16-bit memory block and can be addressed in 16-bit words or in 8-bit bytes.

Two serial RS232C-compatible I/O ports are available on the MEX68KECB. One of the serial ports is used to interface the MEX68KECB to a terminal, required to provide communication between the user and the MC68000. It can range from a high speed CRT terminal to a low speed Teletype. All this is required of the terminal is an RS232C interface. A baud rate generator on the board provides eight clock signals between 110 and 9600 baud. The baud rate of both serial ports is jumper selectable across this range to accommodate varying terminal speeds.

The second serial port is typically used as the interface between the computer board and a host computer. The host can provide mass storage of user generated software as well as more complete software generation capabilities using a resident or cross assembler. The MEX68KECB firmware includes upload and download commands for transferring records between the MEX68KECB and the host. Motorola's EXORcisor and EXORmacs are two potential hosts.

While the serial ports are implemented using two Asynchronous Communications Interface Adapter chips from the M6800 family (MC6850s), parallel I/O and a 24-bit programmable timer are provided by a new Parallel Interface/Timer chip (MC68230) from the 16-bit M68000 family of peripherals. The Parallel Interface and Timer (PI/T) has 24 I/O lines which can function in a variety of I/O configurations.

One parallel I/O port provides a buffered Centronics-compatible printer interface. Hard-copy listings or a complete record of all communications with the terminal can be obtained using this port. Two lines of a second parallel port are designated as an audio tape interface. An audio tape can be used for storage of software instead of, or in addition to, storage by the host computer. The tape format is FSK (frequency shift keying) at a data rate between 1000 and 2000 baud depending on the particular bits sent. Square waves at 2000 and 1000 Hz are used to represent logic ones and zeros. The 24-bit timer generates and measures the period of the square waves. Before being put on tape, information is converted to S-record format, which is a standard data format that includes some error checking.

The MC68230 need not be used exclusively for the printer and tape interfaces. It can be programmed by the user for other I/O and timer functions. Although the MEX68KECB has no expansion bus for connection to other peripheral boards, a small wirewrap area and access to most MC68000 signals as well as to other control signals are provided.

Available at a cost of about $500, the MC68000 Educational Computer Board is unquestionably the biggest basic computer bargain on the market, particularly when considering the capabilities of the MC68000 microprocessor it employs.

BIT BUCKET

ACORN™ COMPUTER SYSTEMS 68-50C

November 20, 1982

'69' MICRO JOURNAL
3900 Cassandra Smith
Hixson, TN 37343

Dear Mr. Williams,

This letter is in response to H. E. Villers' letter appearing in the December issue of 68-MICRO. He is just the type of customer we are looking for !

In January 1982, we decided to support the 68-50 computer with a variety of custom quality boards. In our opinion, the 6809 and 68-50 bus is the best combination for computer experimentation. The 68-50 has some of the finest software and the most reliable hardware on the market.

We purchased a Color Computer with the intention of developing a PROM which would convert it to a monitor for the 68-50 computer. When we studied the circuit, we realized that Tandy Radio Shack had copied it chip for chip from the Motorola handbook. The Color Computer has been described as a 500 hp engine with a 2 qt gas tank. There are four systems on the market with a modified "FLEX" to run on the Color Computer. With the modified FLEX, you have a 500 hp engine with a 2 qt gas tank filled with high octane gasoline.

At ACORN COMPUTER SYSTEMS, we are working on the "RS MISSING LINK"(tm), which will allow a 32K Color Computer to become a part of the 68-50 bus. Now, not only will you have FLEX, but all those expansion boards you read about in 68-MICRO. Our system requires no modification or loss of R.S. warranty and repair service. This method may cost a couple hundred dollars more to begin with, but will be cheaper in the long run. You can add 900k of RAM or PROM with extended addressing, an 80 x 24 video board (keeping the CC video for graphics), a calendar clock, extra ports, a full featured keyboard, voice generator, home control, hard disks, or anything that your heart desires and your pocketbook can afford.

The nice part is that it can be done over a period of years, a little bit at a time. My computer has grown and changed constantly over the past six years. The 68-50 bus, with its many manufacturers, has allowed this personalized expansion.

What we are trying to do, is promote the 68-50 bus through the Color computer. This will bring even more software and hardware to the 68XX system. We hope to have the "RS MISSING LINK"(tm) and our stackable computer modules on the market by March 1983. Write for more information.

Sincerely yours,
Merle Giesfeldt
Merle Giesfeldt, Pres.
A.C.S. Inc.
11931 W. Bluemound Road
Milwaukee, WI 53226

ACS INC
PHONE (414) 257-0300

27 August 1981

MEMORY MAPPING PROBLEMS

Dear Don,

If no one else has, could I bring a small but tricky problem concerning S-BUG to the attention of all 6809 programmers ?

During a routine memory check I noticed that I appeared to have 16 more bytes of memory than I paid for; i.e. I could quite happily read and write the first 16 bytes after the MEMEND address! Rather than being pleased with my bonus I proceeded to check the mother board for address aliasing problems (shorted tracks etc.) After a bit of experimenting I discovered that the fault vanished if I removed my DMAF 2 disk controller board. During the testing of said board I suddenly realised that I was not being very bright. A quick check with the help of Brian Roberts of Middlesex Polytechnic (by the way did you know that they run a multi-machine 6809 cluster system for real time control based on SWTP machines called POLYFLEX ?) confirmed my diagnosis.

When first switched on, SBUG proceeds to locate and remap memory to the correct logical addresses. No matter where your memory is placed, SBUG changes its address to give one 8K block at C000 and the rest as one continuous block from 0000 to MEMEND. What happens if you don't have enough memory to fill all of the available space? What do you find in the 'empty' memory locations? In the old days you'd find 'FF' indicating no memory. However SBUG thinks so much of itself that it fills all the empty memory space with copies of itself! In other words any empty memory blocks are filled with physical address F000-FFFF. If you don't believe me or just want to check, examine X800 and F800, where X is large enough to put you outside your available memory.

This would seem to cause no problem, after all who cares about another copy of SBUG appearing somewhere that you're not going to use. The danger comes from the fact that the DMAF controller board has a physical start address of F000. This means that its 16 DMA address registers appear as the first 16 addresses of every empty memory block! Hence my 'free memory'.

Now just think what a machine code program with a small error that caused it to write to a few bytes just beyond MEMEND could do to your disks!

This problem would of course arise with any peripheral controller card addressed in the F000 region.

Best wishes

Mike James

Mike James
Oak Tree Cottage
Woodhall, Askrigg
Leyburn
N Yorkshire
England

December 3, 1982

68 Micro Journal
5900 Cassandra Smith
Box 849
Hixson, TN 37343

Dear Don:

Poor Motorola seems to be taking a lot of criticism for the MRDY defect in the 6809. In fairness, we should remember that the defect was well documented in the recent data sheets, including identification of the afflicted parts.

We recently discovered that certain AMI brand 6809 chips exhibit a similar problem. I cannot identify the faulty chip sets, but anyone having an AMI 6809 and using MRDY should probably buy a replacement chip, or contact AMI for identification of the bad parts. It is my understanding current AMI parts no longer exhibit the problems.

We are experiencing no difficulties with current Motorola or Hitachi brand 6809 chips.

Sincerely,

Michael Hirsch
Manager, Electronic Systems

ENERGY MANAGEMENT CORP. • 19808 S. VERMONT AVE. • GARDENA, CA 90248 • 213/515-2218

Major George B. STRICKROTH
1613 Maples Avenue
Yuma, Arizona 85364
3 December 1982

Don,

I found Ron Anderson's November column exceptionally enlightening, and of particular interest since I was awaiting the revised edition of Alford & Associates' Scredtor III. As Ron had mentioned in his column, contact with A&A revealed that the update/bug cycle was extending beyond their expectations. I must admit that I was getting frustrated, but my particular frustrations were compounded by an uncooperative UPS. When A&A sent the updated program, it was only to be lost by UPS.

I'd like to report that on 23 November I called A&A with the news that the SCREDITOR III had still not arrived, and they sent one by Federal Express mail, my receiving it on 24 November. I'd also like to very happily report that it's up and running in my SWTPC /09, surpassing my expectations. I am using an Anderson Jacobson 832 RSR daisy wheel printer, and had no problem taking advantage of the proportional spacing, centering, underlining, and bold face capabilities by use of the printer control character functions. After five years with the TSC Editor, it sure is nice to see what I'll have as a final product, and to do true cursor editing.

Keep up the great job with the Journal, and thanks for presenting products like the SCREDITOR III in the advertising.

G.B. STRICKROTH

technical systems consultants, inc.

111 Providence Road • Chapel Hill, North Carolina 27514 • (919) 493-1451 • TWX 510-920-0540

Technical Systems Consultants is proud to announce the release of version 1.01 of the UNIFLEX™ COBOL compiler. This new version has all known errors corrected along with the following major product enhancements:

The COBOL compiler has been restructured to enable it to process considerably larger COBOL source programs. The new compiler should handle programs two or three times as large as the previous version.

INDEXED-SEQUENTIAL files which have alternate keys will now run much faster than before. This was accomplished by a minor change in the internal structure of these files. Because of this structural change, all existing INDEXED files must be copied using the "xsamcopy" utility before they can be used with any programs compiled with the new compiler.

The CALL PROGRAM ... USING verb has been changed so that when the called program terminates, control is returned to the calling COBOL program.

The ACCEPT processing has been completely updated. The new ACCEPT verb will allow the user to enter fully edited numeric fields without the necessity of leading zeroes. Also, the processing of alphabetic and alphanumeric fields has been improved.

The new version of the COBOL compiler is now available using the normal product update procedure.

OMEGASOFT

6809 SYSTEM SOFTWARE

5787 BRANDYWINE CT., CAMARILLO, CA 93010 USA TELEPHONE: 805-987-6426

November 11, 1982

Don Williams
'68' Micro Journal
5900 Cassandra Smith Road
Hixson, TN 37343

Dear Don,

I would like to let everyone know that Omegasoft has moved into its new quarters at :

5787 Brandywine Ct.
Camarillo, CA 93010

Phone (805) 987-6426

I would also like to request that any of Omegasoft's customers that have a compiler version earlier than 2.1a contact me for upgrade information.

Sincerely,

Robert D. Reimiller

Robert Reimiller
Owner, Omegasoft

WINDRUSH
Micro Systems Ltd.

Windrush Limited, (Reg. Office)
North Walsham, Norfolk NR28 9EA
Tel: (06052) 405189
Telex: 97380 SHARET G

'68' MICRO JOURNAL
New Products Editor
P.O. box 849
Hixson
Tennessee
37343 U.S.A.

Your Ref: - Our Ref: WCD/ad 044 Date: 15/12/82

PRESS RELEASE INFORMATION

Windrush Micro Systems Limited announce the immediate availability of their UNIVERSAL 32K MEMORY BOARD for 68-SOC bus 6800 and 6809 computer systems. The product, which retails for 245.00 pounds (excl VAT), has the following features:

- STATIC MEMORY ensures maximum immunity to data errors, does not require halt, wait, or cycle stealing techniques to refresh memory devices.
- Guaranteed 2MHZ operation (tested at 2.2MHZ).
- Extended addressing complies with 68-SOC specifications.
- RAM, EPROM, ROM, or battery backed CMOS RAM devices may be used in any combination.
- Individual device selects enable each 2K device on the board to be removed from the memory map.
- Generous heat sinking ensures cool operation when configured with 32K of EPROM.
- On-board power supply monitor disables battery backed CMOS RAM devices when the PSU voltage falls below a pre-set limit thus protecting the memory from spurious write's as the system shuts down.
- All data and address lines are fully buffered with the latest generation of schmidt input bus receivers to ensure maximum immunity to bus noise and low data error rates.
- Fully socketed.
- Gold plated bus connectors ensure long-term reliability.

B/W Photograph enclosed.

Directors: J.G.Davis W.C.Dick Secretary: V.J.Bell Registered in England and Wales No. 1611025. VAT Reg. No. 372 9531 33

ENDICOTT SOFTWARE
P. O. Box 12563
Huntsville, Alabama 35802
(205) 881-0606

This joystick has been assembled from proven components to provide trouble-free operation.

NOTE: Because of the "tightness" of the new components, the joystick may feel "sticky" in its operation at first. This will disappear after a break-in period, after which you will experience a smooth-acting joystick.

WARRANTY

If any part of the joystick malfunctions or breaks during normal usage within 90 days of receipt of the joyatick, return the joystick prepaid to Endicott Software and we will repair or replace it (at our option) free of charge. Joysticks subjected to obvious physical abuse are excluded from this warranty.

PRESS RELEASE

FROM
Donald J. Sommer
3931 South Burns St.
Seattle, Washington
98118

This is the first announcement for four Network Design and Analysis programs written in Microsoft BASIC for use on the TRS Color Computer. Most of the programs will play on a 16K machine with extended BASIC. They are available for $25.00 each on cassette or 5 1/4 inch diskette. If more than one program is ordered on the same cassette or diskette they are $20.00 for each program.

Program 1 "SNACTIVE" -16k- printer optional
Design Butterworth, Chebychev, or Bessel Low-Pass, High-Pass, or Band-Pass Active Filters. The input is in the fo of break frequency, band width and filter order.

Program 2 "ACTIVE" -32k- printer optional
The design portion is the same as "SNACTIVE". A number of special features have been added to permit screen plotting and outputting frequency response data to the printer.

Program 3 "FILTER" -16k- printer optional
Design Butterworth or Chebychev Low-Pass, High-Pass, or Band-Pass Passive Electrical Filters. The input is in the form of attenuation, break frequencies and input and output impedances.

Program 4 "LADDER" -16k- printer optional
The program calculates the frequency response of general networks which are in the ladder topology. This form includes most passive filters. Some latitude is taken in the definition of ladder because of the large number of different elements permitted. This program is an excellent addition to be used with "FILTER" in the design and analysis of passive R-L-C electrical filters.

Don Williams
68 Micro Journal
5900 Cassandra Smith Road
Hixson, TN 37343

December 2, 1982

I am pleased to announce The Mid-America Color Computer Bulletin Board System, known as 'MACC-NET'.

Located in Kansas City, Missouri, the system is run on a 64K Color Computer, using 3 Shugart disk-drives and an autoanswer modem. The system is up 24 hours a day, except during times of system maintenance and upgrades.

The purpose of this BBS is to provide a forum for Color Computer users to exchange ideas, hints and tricks, problem reports, notes concerning available software and hardware, and as a general sounding board for any aspect of the Radio Shack Color Computer, the Tandy TDP-100, and future Color Computer 'look-alikes'.

A program up-load and down-load feature is available. The program down-load currently has a varied selection of programs. Plans are to include machine-language routines in the down-load, and also text files of additional system documentation, such as ROM addresses, lower memory data areas, etc.

Users of the system can leave messages for each other, leave notices of equipment for sale or wanted to buy. Requests of other users for information, and any information people may have that would be of benefit to others, is always welcome.

An up-load feature has been included, which can be used to send any kind of ASCII data to the system. This can include BASIC programs, Assembler source code, text files of ROM addresses, etc. All data sent to the system will be reviewed before being included on the down-load directory to eliminate duplications. Similar types of informational text may be combined into existing files where appropiate.

The MACC-NET is just beginning operation, and will hopefully grow into a system of immediate use to both beginniers and the experienced user. Any comments on system operation and features are welcome.

The MACC-NET telepone number is (816) 358-MACC (6222). The System Operator is Steve Odneal. Voice number is (816) 356-2345 evenings and weekends, usually until 10pm each night.

Steve Odneal
6600 East 73 Terrace
Kansas City, Missouri 64133

December 15, 1982

Computer Publishing Center
'68' Micro Journal
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343

Gentlemen:

The first 68000 based machines are beginning to appear. Recently, we took delivery on an Altos ACS68000 system with a 40mb disk able to handle sixteen users running Bell Lab's Unix Operating System. The cost was less than $1000/user.

One of the first things we did was to run some benchmarks. Of particular interest was the sieve program from Vol. IV, No. VII, July, 1982. (Also see No. IX.) The 68000 ran this program in 6.3 sec., compared to 22 sec. for Intrel C on the 6809 and 15.5 sec. for Whitesmith C on the Z80. However, this is slow compared to the 2 sec. it takes to run on a $30,000 Zilog System 8000!

There are some offsets. As a practical matter, most programs tend to be I/O bound, and here we find the Altos to have disks that are 2.5 times as fast as the Zilogs. In particular, to do 1000 sequential writes followed by 1000 sequential reads and 1000 random reads of 512 byte records takes the Altos only 21.5 sec. The Zilog takes 52.4 sec. This performance results mostly from the use of eight inch Quantum drives as opposed to the much slower 5.25 inch drives others are using. The message is clear. With Unix, the disks are more important than the chip.

Returning to CPU bound processes, the particular implementation of the language and operating system can be important. Zilog has had Unix long enough to have optimized the C compiler for their hardware. Altos is still using the standard version. Zilog keeps as much of the stack as possible in registers. This is not standard practice, and in fact, can hinder pointer manipulations. On the other hand, it can make the system very fast. Finally, the 68000 is presently too fast for the available memory management and RAM chips, so wait states are used on the Altos, further slowing things down. When Altos gets their compiler optimized, obtains faster chips and moves to the 12.5 Mhz CPU, their CPU bound processes should be competitive with anyone's. In the meantime, the overall performance of this system seems to match that of machines costing five times as much.

Rumor has it that a smaller machine, which will handle about five users and compete in the $5-10,000 market is on the way next Spring. An 8086 model has already been announced, so the 68000 model should not be far behind. This machine should be very attractive for the smaller business or the larger home.

Sincerely,

Robert B. Peirce

Robert B. Peirce
121 W. Edgewood Dr.
McMurray, PA 15317

SPECTRUM PROJECTS
93-15 86 DRIVE
WOODHAVEN,N.Y.11421

Voice line (212)441-2807

Data lines (212)441-3755/3766

How To Upgrade A TDP100 To 64K
By BOB ROSEN of SPECTRUM PROJECTS

The TDP100 has a different RF shield cover from the Color Computer. It only covers the 6883 and the 8 memory chips. You have to go underneath the board and straighten out 7 tugs that are bent parallel to the board. You can then remove the RF shield. There are 3 jumpers to move from 16K to 64K. Two of them are to the left of U21 and the other is above U20. Also you have to add a jumper that is to the left of U17. Then remove capacitors C58,C60,C62,C64,C66,C68,C70 and C72. That's it! there are no jumpers to hook up and pins to bend up to access the full 64K. Modifications in this new board (only on TDP100 computers) helps eliminate those extra steps. Some other observations about this new board. It uses a new chip for the keyboard PIA. The 6822 was added to eliminate hash during interrupts. The chips have different numbers than found in the Color Computer Technical Manual. The two PIA's are now U17 and U18 instead of U4 and U8. There is also a better relay for the cassette than was used previously. My final comments is that this board is easier for upgrades and better designed than the series "D" and "E" boards.

109 Carolyn Avenue
December 3, 1982

Mr. Don Williams
'68' Micro Journal
P.O. Box 849
Dixon, Tn. 37343

Dear Mr. Williams:

I need some help. I need routines to initialize my MP-S2 at E000 and list a program to my Microline 82A printer, also I would like to run a program to the printer using TSC cassette basic.

My system consists of the following:
1. SWTPC, 69/A with 56K ram, MP-S2 serial interface with A port at E000 and B Port (control) at E004, JPC TC-3 interface at E040 with Motor control and CPM/3 at F000 in 8 prom running most of the time at 4800 BAUD with several programs at 9600 BAUD (without the TC-3 I would have given up this SWTPC system soon after I bought it).
2. Fisher CR-110 stereo cassette deck.
3. Metronics professional keyboard, video display board, and modulator.
4. RCA television.
5. Okidata Microline 82A printer.

I have tried everything I can think of to make this printer work with the MP-S2. I even sent the MP-S2 board back to the factory and asked for some help with routines to make it work, they did not acknowledge receiving the request. They said there was nothing wrong with the interface. I also sent the Microline 82A back to the factory which was returned in eight days. I figured it would be some time before receiving nything back from SWTPC. Being in a hurry to try the printer, I bought a Commodore Computer. There was no problem getting the printer to work with the Commodore, and I was pleased.

The MP-S2 came back one month after I sent it in. I have been unable to make the two work together. I even wrote TSC for help, they sent driver routines to be used with Flex, that is all.

I have the printer now running on 8 bit word, 2 stop bits, and no parity. This is the way I would like to use it with the MP-S2 and TSC basics.

I would appreciate help in this matter and would be glad to pay for it. I also have a local friend looking for help. He has a SWTPC 6800 unit and wants an 8K basic by SWTPC on Kansas City Standard tape. He sent a check to SWTPC but they returned his check saying the tape basic is no longer available. Where may we find this basic and what is the price?

It appears to me that some companies are so called upgrading themselves right out of business. Most of us have to start with basic units at reasonable price and expand from there. We need lots of support from manufacturers. If we cannot get help from the manufacturer we will go elsewhere. A company will cease to exist, let alone grow without public concent.

Thank you Don.

Sincerely,

Dewey H. Noble, Jr.
109 Carolyn Avenue
Salisbury, Maryland 21801
(301) 742-6702

THE 6809 COMPANION

Dear Don,

I am writing to tell you of my new book "THE 6809 COMPANION". Although there are one or two books around that deal with the 6809 now, when I started work on the project, six months ago, there were none! I have always been a keen 68 user and was one of the first to have an SWTPC 6800 system in the UK back in the good old days when computers were real computers (i.e. mostly electronics and MIKBUG!). Along with others I slowly worked my way up the tree from cassettes and 4K BASIC to Disks, FLEX and a CIMIX box. When the 6809 arrived I was pleased to discover that I had not been wrong sticking with the S50 Bus. And the use of an S100 machine at work also helped to convince me that I was right!

After writing a few articles for some of the UK computer magazines on the S50 and the 6809 I decided to write a short book on explaining some of the software features of the 6809. After spending so long converting many of my own 6800 programs to the 6809 you can understand why this was the first chapter I started work on. Motorola may say that conversion is easy but there are many things that can go wrong that Motorola never thought of. Believe it or not, this is the chapter I personally most use!

The rest of the book deals with the sorts of topics you'd expect in a machine companion:- history, architecture, addressing modes and the instruction set (fully commented). A chapter on programming style tries to point out the strengths of the 6809 and how to make use of them. Interrupts are also dealt with from the point of view of the programmer in yet another chapter.

Although I set out to write a book for programmers, I felt that a chapter on hardware was necessary. All too often have I come across 6809 users (especially in industry) under the impression that hardware comes only from Motorola! So I have tried to give an impression of the S50 bus and what's available, including system software. (I have to admit to being a FLEX freak!)

To cap it all the whole book was produced using an SWTPC 6809, TSC's text editor and typeset using a 6809 program I produced specially.

After all this I wanted to find a publisher who could market my book for a reasonable price and, having prepared my photo-ready copy, my search led me to Babani Publishing. The result is a compact volume costing only £1.95 (approximately $4.00!) which is cheap enough to buy a new copy when the old one falls apart from over use.

Mike James

Best wishes

Mike James
Oak Tree Cottage, Woodhall
Askrigg, Leyburn
N Yorkshire, England.

PRESS RELEASE
COMMUNIQUÉ DE PRESSE
PRESSEINFORMATIONEN

ISSUED ON BEHALF OF: swtpDolpr/DB
Southwest Technical Products
Unit 12
Trenham Road
Orton Southgate
PETERBOROUGH

FOR IMMEDIATE RELEASE

DoI BACKS COMPUTER SYSTEM FOR DOCTORS

The Department of Industry is to finance up to four of the new ARIES 20/256 GP computer systems for General Medical Practicioners under its "Support for Innovation" scheme

It will the first DoI sponsored system to be installed in a GP's surgery as well as the first multi-user GP system to be backed by DoI.

The ARIES 20/256 uses the latest SWTP 8 Plus computer, running the UniFLEX (UNIX-like) operating system with 256k RAM. Backing storage includes a 20 megabyte 5.25 inch Winchester disk, a 20 Mb tape cartridge archive unit and a 1.3 Mb 8 inch floppy disk drive. Two intelligent VDUs and a 240 cps printer are being provided with each system. The A+ can take up to 12 terminals.

The ARIES GP software has been specifically designed to computerise administration in the GPs surgery. It includes programs for age/sex registers, preventive medicine, patient recall, repeat prescriptions, drug use analysis, patient history taking (MICKIE) and word processing.

The first ARIES 20/256 system is to be installed at the Chiswick Health Centre, London W4, just 75 yards from the offices of ARIES Information. These Health Centres are a five-partner practice with over 13,000 patients.

The GPs participating will also be involved in the valuation program for the "Micros for GPs scheme", also backed by the DoI.

The four systems will cost the DoI £80,000.

For Further Information Contact:

Tim Swann, ARIES Information Ltd.
10 Harley Hse Marylebone, London, NW. Tel:- 01-994 6477
Nick van Basten, Intermark PR Ltd. Tel:- 01-637 8782/3
John Burnett, SWTP Tel:- 07333 234433
Mike Hitchcock, DoI Press Office. Tel:- 01-212 5484

Issued by: INTERMARK Public Relations Ltd.
38 Charlotte Street London W1P 1HP Tel: 01-637 8782

GENTLEMEN;

PLEASE FIND ENCLOSED CHECK IN THE AMOUNT OF $ 48.50 IN PAYMENT FOR A 3 YEAR EXTENSION OF MY PRESENT SUBSCRIPTION. I HAVE ENCLOSED A COPY OF MY MAILING LABEL TO ASSIST YOU IN THIS REGARD.

I ENJOY READING THE JOURNAL AND I HOPE YOU CONTINUE TO BE VERY SELECTIVE IN THE ADS WHICH YOU WILL ACCEPT.

I ONLY HOPE THAT THE MANUFACTURERS OF 68000 EQUIPMENT REALIZE THE IMPORTANCE OF STANDARDIZATION OF BUS AND OPERATING SYSTEM SO THAT WE AS CONSUMERS ARE NO LONGER A SPLINTER GROUP. THE DE FACTO STANDARD OF THE FUTURE WILL BE THE VME BUS AND A UNIX OPERATING SYSTEM. THIS WILL ALLOW 68000 USERS TO UPGRADE FROM 24 BIT ADDRESSING TO 32 BIT ADDRESSING WHEN MOTOROLA ANNOUNCES THE ENHANCEMENT. IT IS MY FEELING THAT MORE PROGRAMS AND BOARDS WILL BE AVAILABLE TO THE USER IF THEY ADHERE TO THE VME BUS AND UNIX. THE 68000 IS THE MICRO OF THE FUTURE. THE SYSTEMS USING THE 68000 CAN BE SIMPLE OR COMPLEX AS THE USER DESIRES. THE COST OF THE 68000 CHIP ITSELF HAS DROPPED CONSIDERABLY IN THE PAST YEAR. THERE ARE A NUMBER OF SECOND SOURCES FOR THE CHIP AND SEVERAL OF THESE SECOND SOURCES ARE PRESENTLY DEVELOPING PERIPHERAL CHIPS WITH MOTOROLAS BLESSING. THE TECHNOLOGY WILL BE SHARED WITH THE OTHER SECOND SOURCE PARTIES. THE COSTS WILL BE DROPPING ALSO.

THE VME BUS WAS A CONCERTED EFFORT BETWEEN MOTOROLA AND THESE SECOND SOURCES WHICH INCLUDE SOME FROM OVERSEAS. MOTOROLA WILL SUPPORT BOTH THE VERSABUS AND THE VME BUS. THE VME BUS USES A PIN AND SOCKET ARRANGEMENT TO CONNECT BOARDS TO THE MOTHER BOARD. NOT COMPATIBLE WITH THE S550 OR S564 BUT IT IS FAMILIAR. THE CARD SIZES ARE STANDARD EUROCARD DIMENSIONS. CARD PRICES WILL COME DOWN AS MORE MANUFACTURERS START PRODUCING CARDS THIS YEAR. OUR "OLD" 6800 AND 6809 MICROS DO NOT HAVE TO BECOME OBSOLETE. THEY CAN BE UTILIZED AS PERIPHERAL CONTROLLERS. MULTIBUS,S564 BUS SYSTEMS WILL SLOWLY BECOME UNPOPULAR AND THEREFORE THEIR CARDS WILL BE MORE COSTLY. AT LEAST ONE FAMILIAR MANUFACTURER IS KEEPING A CLOSE EYE ON VME DEVELOPMENTS (SMOKE SIGNAL BROADCASTING). THERE MAY BE OTHERS LOOKING BUT RELUCTANT TO TIP THEIR HAND AT THIS TIME. WE AS CONSUMERS MIGHT GO WELL TO DELAY A BIT IN COMMITTING OURSELVES.

PITY THE POOR S100 BUS USERS. THEY WILL HAVE TO START ALL OVER IN ORDER TO JUMP ON THE 68000 BANDWAGON. THEIR BUS WILL NOT SUPPORT THE HIGH DATA TRANSFER RATES.

SINCERELY YOURS,

LARRY WEAVER

LARRY WEAVER
6004 OAKWOOD DR.
WICHITA KS 67208

COMPUTERWARE™
Microcomputer Sales and Software

Computerware™ introduces its SPACE AMBUSH on cassette and disk for the Radio Shack Color Computer and TDP System 100.

You're in a crater's valley on the planet Orgath doing routine surveillance, (your job as commander of the Galaxian Protector Fleet) when a cloud of marauding terrorist ships ambush your station. Their first attack has left you with no vertical boosters, limiting your movement to surface manouvers. Galaxy warfare treaties restrict your weapons to only the short-range phasers. But that's enough to destroy any attacking space craft that descends close enough to drop bombs.

The enemy gang is a collection of aggressive hoodlums from throughout the galaxy using a varied collection of hijacked crafts including transports, blinkers, bombers, flippers, spinners, and even flagships.

CLASSIFIED ADVERTISING

CLEARANCE SALE: COMPLETE SWTPC SYSTEM INCLUDING 69/A COMPUTER WITH 56K, DUAL SERIAL I/O AND PARALLEL I/O. TWO DUAL DENSITY 5 1/4" DRIVES WITH QUAD-DENSITY CONTROLLER THAT WILL HANDLE FOUR DRIVES. CT-82 TERMINAL, CONNECTING CABLES AND ALL MANUALS. SOFTWARE INCLUDES LATEST FLEX (2.8:3), TEXT EDITOR, ASSEMBLER, TEXT PROCESSOR, BASIC AND XBASIC. FIRST $1450.-TAKES IT ALL.
MARTY PETERSON, 327 ELM ST., QUINCY, ILL 62301, (217)222-8200 BUSINESS HOURS.

***

$475. ALL- 80 X 24 W/PROG CHAR,48K W/8K INSTALLED 4K RAM(2 EACH), 2708 PROG & 2 EACH 8K EPROM, 6800 CPU. MORE W/SOFT.
LABOMARD, 1541 SATURN BLVD 207, SAN DIEGO, CA 92154

***

SWTPC: MP-R $60; MP-S2 $75; MP-L2 $75; MP-N $45; MP-LA $25; MP-32 $125; 6809 + 8KB IMPROVED POWER $300; GIMIX: 32KB $200; PERCOM: CIS-30 $50; REASONABLE OFFERS ACCEPTED.
603-774-7762 TIL 10PM EST, MANFRED

***

SWTPC 6800 W/24K, MP-C, MP-S, MP-LA, SWTBUG, CT-1024 TERM., AC-30, GT-6144, JOYSTICK, PR-40, DOCUMENTATION. $350.00 LEEDEX 12" MONITOR AVAILABLE.
WILLIAM TANT, 1723 CO-AH-BAR ST., ROCKY MOUNT, NC 27801, 1(919)977-0322

***

ARCADE BOARD WITH F-BASIC $300., HAZELWOOD VC256 $200.
DAVE HANON, DAYS (615)698-3337

***

MEK680002, 2 M68MM01, 1 MMS68104, 4 MEX68WW, 2 MOTOROLA CAGES W/BACKPLANE. $200.
S.D. SINKLER, 100 PINETREE RD, RADNOR, PA 19087, (215)687-1044.

***

FOR SALE: SWTPC 8K STATIC MEMORY BOARDS, $35; THOMAS INSTR. 24K STATIC, $70; THOMAS INSTR.. 48K RAM/ROM WITH 24K, $150. ALL 2MHZ.
STEVE SHERMAN 303-666-6058, EVENINGS.

***

TEN MOST-ASKED QUESTIONS

ABOUT DYNACALC™

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

1. **What is an electronic spread-sheet, anyway?**
   Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.

2. **Is DYNACALC just for accountants, then?**
   Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. There's even a built-in sort command, so you could use DYNACALC to manage small data bases - up to 256 records.

3. **What will DYNACALC do for ME?**
   That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalc, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications.

4. **Do I have to learn computer programming?**
   NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Built-in HELP messages are provided for quick reference to operating instructions.

5. **Do I have to modify my system to use DYNACALC?**
   Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.

6. **Will DYNACALC read my existing data files?**
   You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, RMS data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.

7. **How fast is DYNACALC?**
   Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.

8. **Is there a version of DYNACALC for MY system?**
   Probably. You need a 6809 computer (32k minimum) with FLEX or UniFLEX operating system. A version for OS-9 is also in the works. You also need a decent CRT terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UniFLEX version of DYNACALC also allows you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX and Data-Comp's F-MATE. A version for Frank Hogg's Color Computer FLEX is also being done.

9. **How much does DYNACALC cost?**
   The FLEX versions are just $200 per copy; UniFLEX version $395. Foreign orders add $10 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.

10. **Where do I order DYNACALC?**
    See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 a.m. to 6 p.m., Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX versions. Software serial number is required for the UniFLEX version of DYNACALC.

ORDER YOUR DYNACALC™ TODAY

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 7A Burton St., Darlinghurst, NSW 2010 Sydney. Telephone: 02-357-5111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

CSC

Computer Systems Center

13461 Olive Blvd.

Chesterfield, MO 63017

(314) 576-5020

UniFLEX software prices include maintenance for the first year.

DYNACALC, DYNAMITE, and DYNAMITE + are trademarks of Computer Systems Center.

F-MATE is a trademark of Data-Comp.
VisiCalc is a trademark of VisiCorp.
OS-9 is a trademark of Microware and Motorola.
FLEX and UniFLEX are trademarks of TSC.

ALSO FROM CSC

DYNAMITE +

"THE CODE BUSTER"

now available for UniFLEX
OS-9 version soon

DYNAMITE + is a new version of DYNAMITE, our popular 6809/6800 disassembler package for 6809 FLEX. Present users of DYNAMITE can upgrade to DYNAMITE + by sending us the original DYNAMITE diskette and $40 (plus $5 for foreign postage). DYNAMITE + does everything DYNAMITE does, and more! A cross-reference generator has been added, label files are now maintained only in text form (LABEL EQU $xxxx), and boundary file specifications have been tremendously simplified, which makes it easier to disassemble large programs containing lots of big tables.

The UniFLEX version of DYNAMITE + does everything the FLEX version does, and also automatically handles system calls and 'info' areas.

DYNAMITE + is available for $100 per copy on FLEX (specify diskette size), and $300 on UniFLEX. Foreign orders add $5 per copy for postage.

SCREEN EDITOR/WORD PROCESSOR FOR DOS6x, FLEX, OS-9

IF YOU are CONSIDERING UPGRADING YOUR EDITING AND PROCESSING CAPABILITIES, HERE ARE A FEW GOOD REASONS TO CONSIDER SCREDITOR...

REASON #1 - SCREEN EDITING

Screen editing (sometimes called cursor-based editing) is the most efficient way to enter your thoughts into the computer. Since you see the actual text you are correcting on the screen as if you were looking at a printed page, organization is a matter of manipulating rather than guessing.

REASON #2 - DYNAMIC SCREEN FORMATTING

With dynamic screen formatting, the screen becomes an image of what the completed job will look like. Within the capabilities of the display terminal itself, you are constantly looking at what you would have if your job were printed. Every change you make, every addition or deletion, is automatically incorporated in a fully-justified, ready-to-print form. You no longer have to guess what all those cryptic processing commands you may have used with other software will turn your finished job into!

REASON #3 - NO MACHINE LANGUAGE PATCHING

SCREDITOR III is unique in that you NEVER have to make machine language patches to integrate it into your system. If you can answer a series of questions about your printer and terminal's capabilities, installation is complete.

REASON #4 - NO EXTRA "PRINTER PROGRAM" NEEDED

SCREDITOR III is complete as purchased. You get screen editing, formatting, dividing lines merging and much more, all in a single program. And it can through the competition and provide many of the features of SCREDITOR FOR III, how much will you pay? And, more important, how many programs will you have to buy?

REASON #5 - OPERATING SYSTEM SUPPORT

To our knowledge, SCREDITOR III supports more processors and operating systems combinations than ANY other single editing and word processing package available. OS-9 (level one and two), Flex and SSB DOS, 6809 and 6800 versions are available. Except for those commands which are peculiar to a particular operating system, SCREDITOR III is identical in operation in all versions.

REASON #6 I/O - COMPATIBILITY PROBLEMS SURROUND YOU

If you purchase another company's product for your FLEX-9 system and decide to run OS-9, how much will it cost you for that same copy of their software if/if it's available. How finished PO. I with SCREDITOR III, tell us you are upgrading and you pay ONE of the price you paid for the first copy and you can have the same great capabilities with both operating systems!

REASON #7 - CAPABILITIES

Not only is SCREDITOR III the only COMPLETE editing and word processing program available, it's also the most powerful. Some of the other word processing programs available give you large multi-column editing. Or allow you to add and delete a single row and columns of numeric data. Or allow you to change margins and tabs at any time during editing just like a typewriter. Or give you commands to re-justify and change the width of paragraphs and lines on-screen, while you watch. Or let you add re-execute commands. Or allow automatically calculated mixed text and command entry (sophisticated a single keystroke (macros). Or let you move all of those macros to a disk file to be re-loaded for use in later edit sessions. Or edit your system. Or...

REASON #8 - DOCUMENTATION

SCREDITOR III has the most comprehensive documentation available for any 6809 editor/word processor. The Technical Reference Manual is currently about 300 pages long. The Training and User's Guide is and we are now preparing will be about another 300 pages. We're also working on our new Tutorial Cassette (60+ hours of instruction). We even provide a HELP command which gives you the syntax of all of SCREDITOR III's commands and at least one instruction of description of each!

REASON #8 - PRICE

Considering its capabilities, SCREDITOR III is the least-expensive program of its type available today. Period.

REASON #9 - HAPPINESS

We want your business. Think how happy you will make us if you decide to buy SCREDITOR III!

The following information should answer most of your questions about SCREDITOR III. If you have additional questions, please call during our office hours, or, better still, order the manual (when you decide to buy SCREDITOR III, we'll apply your manual purchase against the software price).

PROCESSORS SUPPORTED - 6800, 6809

OPERATING SYSTEMS - FLEX, SSB DOS, OS-9 (level I and II)

DISPLAY REQUIREMENTS - Serial terminal with computer-addressable cursor. 4800 baud minimum data rate, 9600 or better preferred. Clear EOL and clear EOS not required but highly desirable. Insert and Delete line improve display speed. Suggested terminals include Televideo, Heath/Zenith H-19, SWTPCo. Virtually all addressable-cursor terminals can be used. ANSI terminals supported by the time you read this.

Memory-mapped displays require software to provide terminal look-alike capabilities. Sample software included with FLEX and DOS6x orders.

Minimum display size is 16x64. Maximum is 255x255. No software patching is required to adapt software to display size.

Unique configurations may be supported by calls to external user-supplied sub-routines.

KEYBOARD REQUIREMENTS - ASCII character set with control codes required. Function keys generating multiple-character sequences also supported. Control code and function key assignments user-defined and changeable at any time without patching.

KEYBOARD INTERFACE - PIA parallel, ACIA serial, or external keyboard handlers. Interrupt-driven operation type-ahead type-ahead is a standard user-selected option.

PRINTER HARDWARE - Current version supports standard printers. Specialty printer capabilities controllable by transmission of control or escape sequences imbedded in text. Underline, boldface, double width simulated on printers with backspace capability. Proportional printing available soon.

PRINTER INTERFACE - Program uses existing user-supplied software to transfer characters from SCREDITOR III to printer hardware by calls to user sub-routines.

COMMAND SET - Fourty-six commands (or more by the time you read this) with multiple options on most commands. Thirty-one single key commands for often-used operations such as cursor movements, inserts, deletes, paragraph formatting, etc...

OPERATION AND DISPLAY - Text entry by 'window-type' screen/cursor editing. Commands by single-key or line entry. Continuous display of current tabs/margins, edit modes, cursor row and column, current line number, horizontal display range.

REQUIRED MEMORY - (program memory) - Program size - approximately 20 kilobytes. Minimum memory for useful editing - 32 kilobytes.

DISK MEMORY - Configuration files (3 minimum) - 1 to 2 sectors each
HELP file - disk-resident, approximately 10 kilobytes

********** AT LAST! **********

AVAILABILITY - Please allow one to two weeks for delivery. Disks are built as orders are received to allow the most current possible revision of software to be shipped.

PRICE - SCREDITOR III manual, disk $150.00
manual only (price may be applied to software order) . . . . . $ 40.00

IMPORTANT INFO

We accept Mastercharge and VISA credit card orders, personal and business checks drawn on U.S. banks and in U.S. dollars, and COD's to the continental United States. Domestic orders shipped by United Parcel Service unless otherwise requested. Overseas orders shipped by air parcel. COD's shipped by United Parcel Service only. Open accounts to government agencies, educational institutions, established accounts and Fortune 1000's only. Virginia residents add 4% state sales tax. On orders under $100, add $3 for shipping and handling. On overseas orders add $10 for shipping.

THE BEST NEWS!

In our haste to do business, too often the temptation to compromise our faith weighs heavily. Even though we have forgiveness when we confess our sins from a God who loves us, we would like to take a moment to ask forgiveness from any of our customers whom we have slighted. Orders do get lost or mis-shipped, and mistakes do happen. If we fail to completely please any of you in either our responses to your order, inquiries, or for whatever reason, let us know. We will do anything in our power to satisfy your problems. We thank all of our customers for your prayers, and we pray for each of you that you will come to know the love of God in your lives that we have found!

ALFORD & ASSOCIATES
P.O. BOX 6683
RICHMOND, VA., 23230
804-320-6722

CoCo SPEECH SYNTHESIZER

The Color Computer is one of the best things to happen to computing since the SS-50 bus! At the price, there is no better way to get into personal computing, and we are now offering a speech synthesizer for the CoCo! With most of the same features of our VS-1, the SP-1 can give your CoCo the power of unlimited speech. Unlike software approaches to speech, the SP-1 does not eat your memory, and provides the clearest, easiest to understand speech we have heard!

The SP-1 is a completely assembled and tested unit which plugs directly into the ROM-PACK slot on the side of the computer. Unlike the VS-1, however, no external speaker is required, as the SP-1 routes its audio directly through the computer and into your television monitor!

We also have a version of our VOX-EDITOR to let you build your voice messages quickly and easily, and then save them to cassette when you are done. The speech files can later be read by Basic in your application program! The comprehensive manual we supply gives you all the information you need to make your CoCo talk!

SP-1 pack and manual $179.95
SP-1 manual only (price applied to SP-1 purchase) . . . . . $ 10.00

TRANSLATOR FOR SP1

Our SC-1/COLOR translator uses the same highly accurate algorithm as our SC-1 for the VS-1, but drives the SP-1 on the color computer. The SC-1/COLOR, however, is even easier to use! When the SC-1/COLOR translator is loaded, it becomes a part of Basic! To translate and speak English text, numeric data (even punctuations if you desire!), you simply program PRINT #1 followed by your text. The translator takes it, converts it to a phoneme stream, and drives the SP-1 directly!

SC-1/COLOR cassette and manual (requires SP-1 for operation) $29.95
(manual not available separately)

SSB UTILITIES DISK

Two of the shortcomings of SSB DOS are that, one, with six-character file names and default extensions it's hard to know from a name what is in a file, and, two, with no disk information sector the only way to know what is on a disk is to list the files! Since we couldn't conveniently make the file names longer, we did the next best thing and wrote a family of programs to allow you to "name" your disks!

Utilities #1 provides a NAME utility to build a disk information sector on each newly-formatted disk. UPDATE lets you change this sector. LOCK lets you write, delete or even LIST protect the files on the disk. LIST is a better listing utility which reads the disk access code from the information sector and will only list LOCK'ed files if you supply the access code. A number of other programs are also included on the disk.

The manual provides information about the way SSB disks are formatted and how the chained sector method work in addition to complete descriptions about each of the programs provided on the disk!

UTILITIES #1 disk and manual $29.95
(manual not available separately)

FANTASTIC PROOFREADER

We are proud to offer Pete Stark's Spell 'n Fix proofreader. We examined a number of very good pieces of spelling correction software. We compared ease of use, speed, correction capability, dictionary maintenance and expansion , disk space overhead, etc..., and found that Pete Stark of StarKits has done a remarkable job on this program. Look at these features and make your own decision,...

MULTIPLE DICTIONARIES - Spell 'n Fix allows you to create multiple dictionaries. In this way, if you want, you can put technical and special words in a separate dictionary. Then, when you proofread your documents, select the dictionaries which have the words you need, thus speeding up the job!

EXECUTE DOS COMMANDS - Need to list your disk to decide which dictionary or text file to proofread? Want to delete an old file? View a file? Most of your DOS commands can be executed directly from within Spell 'n Fix.

MULTIPLE CORRECTION OPTIONS - With Spell 'n Fix, you can add new words to your dictionary at any time. Or, if you want, you can create a new dictionary for words not found in the existing dictionaries. Or simply print out the new words. Or even ignore them! CORRECTION OR MARKING - With Spell 'n Fix, you can mark incorrectly-spelled words in your text for later correction, or you can tell Spell 'n Fix to correct them as they are found. The choice is yours!

COMPRESSED DICTIONARY - Spell 'n Fix uses either of two forms of dictionary, and converts either to the other at your command. The first type is a fully expanded dictionary with one word per line, allowing you to use your editor to make corrections, deletions and additions easily. The second type incorporates a slick compression method to dramatically reduce dictionary size and increase access speed.

After looking at our options, we decided that this was the program for us, and are proud to be the exclusive agents for StarKits for the SSB DOS68 and DOS69 versions. We can also supply versions for 6800 and 6809 Flex. The OS-9 version will be available soon directly from StarKits. StarKits also has a version for the Color Computer! At the price, we feel that no other program can do what Spell 'n Fix can....

SPELL 'N FIX disk (with 17000 word dictionary) and manual $89.28
(manual not available separately)

SS50 SPEECH SYNTHESIZER

The VS-1 is a completely-assembled, tested unit for the SS-50 I/O bus which gives your computer the power of unlimited speech. Since the VS-1 is programmed phonetically, there is essentially no limitation on what you can make your computer say! The VS-1 programs just like a parallel printer driver, and requires an average of only one phoneme per letter in English for a very low message memory storage overhead. The data rate is typically a low 70 baud.

The VS-1 is supplied with the easiest-to-use, most flexible speech editing program available. This program makes building your speech files or data a real pleasure. We also supply additional software and sample speech files on the disk to give you a 'leg up' on making your computer talk.

The manual provides sections on speech generation theory, operation of the supplied software, complete technical descriptions, a sample speech dictionary and phonetic code charts, fifty pages in all!

If you are a roll-your-own nut, we provide the bare board or manual or disk or synthesizer chip, or combinations (We suggest that only VERY experienced persons attempt such assembly due to the fact that we do not provide assembly instructions beyond parts list parts layout and schematic).

The VS-1 is available for 6800 and 6809 FLEX, 6800 and 6809 SSB DOS, and OS-9 level I and II.

VS-1 complete with manual and disk $164.95
VS-1 manual and disk only (may be adapted for brand-X) . . . . . $ 29.95
VS-1 manual only (may be applied to VS-1 order. Look before you leap!) . . . . . $ 10.00
VS-1 bare board (build your own...requires experience!) . . . . . $ 19.95
SC-01A synthesizer chip . . . . . $ 60.00
VS-1 bare board, chip, disk and manual . . . . . $ 99.95

TRANSLATOR FOR VS1

If you have our VS-1 (or even brand-X), and want an easier way to make your computer talk, our SC-1 translator is for you. The SC-1 takes plain English text from a disk file or your keyboard, translates it to a phoneme stream, and sends the resulting output to either the speech synthesizer hardware or to a data file for later use.

The SC-1 translator is one of the most accurate such programs we have seen, making typically only one 'mistake' per sentence, for an average accuracy of about 95%! Due to the type of algorithm we are using, even proper names are generally translated correctly! Where the SC-1 is being used to compile "canned" messages for later use, we provide an additional program which takes a binary phoneme file created by the SC-1 and converts it to a speech file which may be edited by our VOX-EDITOR (supplied with the VS-1). In this way, you can quickly create your messages and then "fine-tune" them!

The SC-1 translator is available for 6800 and 6809 FLEX, 6800 and 6809 SSB DOS, and OS-9 level I and II. If you are using brand-X, order the SC-1 with the VOX-EDITOR on the same disk!

SC-1 disk and instructions $39.95
SC-1 disk, instructions with VOX-EDITOR $49.95

TREK-68

Our TREK program is the best 'real-time', 'live-action' game available for 68xx computers using memory-mapped displays. This program does not require graphic capabilities from your display. Otherwise, this is a full arcade-type game. No triviality here, either. The suicide option has only been won by a handful of people world-wide, and can run for hours!

TREK is available for 6800 and 6809 FLEX and 6800 and 6809 SSB DOS, and requires both a memory-mapped display and a source of 50 or 100 Hz interrupts (such as the MP-T timer board).

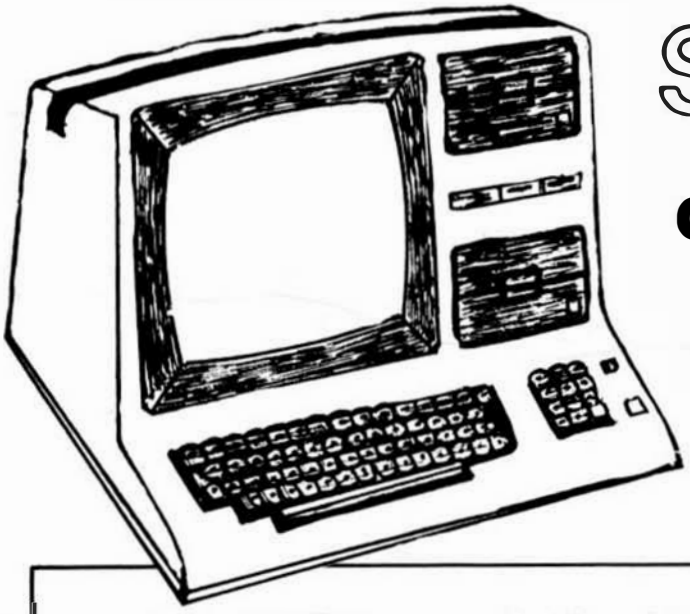
TREK disk and manual $24.95

COLOR-TREK

Like our other version of TREK, COLOR-TREK provides you seat-of-the-pants action. This program uses no routines from Basic, and so can run on non-extended Basic machines. 16k memory is required. Only one player to our knowledge (other than the author) has won the suicide option of this game, and it took him fifty-seven minutes! No triviality here!

COLOR-TREK cassette and manual $17.95

Universal Data Research, Inc. Introduces



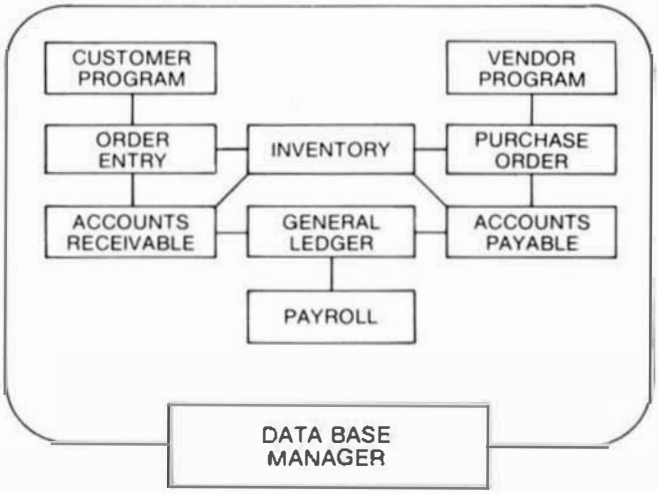
SOFTWARE
for the COLOR COMPUTER and the TRS 80 Model III

- Data Base Manager ____ $150
- Church Contributions Package ____ $150
- Single Entry Ledger ____ $ 95

FLEX* and UniFLEX* SOFTWARE for the 68XX Operating System . . .

INTEGRATED BUSINESS PROGRAMS for large and small companies

	FLEX	UniFLEX
Accounts Receivable	$395	$495
Accounts Payable	$395	$495
General Ledger	$395	$495
Inventory 2	$395	$495
Payroll	$395	$495
Data Base Manager	$450	$550



STAND-ALONE SOFTWARE

- Single Entry Ledger
- Check Balancer
- Costing and Estimating
- Church Contribution Pkg.
- Church School Billing
- Fund Raising
- Custom Programming
- M.F.I. (Extensive Mfg./P.O.)
- Bulk Mailings
- Maintenance Schedule
- Vendor Program
- Purchase Order
- Accounts Payable
- Accounts Receivable
- Customer Program
- Order Entry

UNIVERSAL DATA RESEARCH INC.

*FLEX & UniFLEX are Trademarks of Technical Systems

2457 Wehrle Drive,D-1, Buffalo, New York 14221
Phone (716) 631-3011

master charge VISA

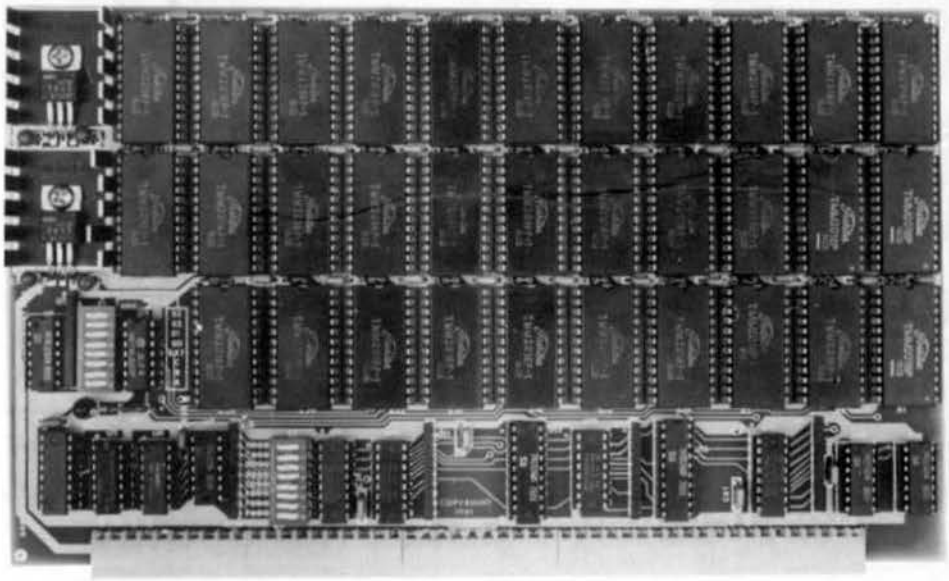
64K SS-50 STATIC RAM

NEW!

$219.00 (48K KIT)

NEW!

LOW POWER!



RAM OR EPROM!

BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS -	$18.00
FULL SOCKET SET -	$15.00

ASSEMBLED AND TESTED **ADD $40**

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K Kit	$269
64K Kit	$319

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $2.00 postage. We pay balance. Order under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

ARCADE 50

POWERFUL COLOR GRAPHICS
Uses the new TMS9918A Video Display Processor. High resolution 256 x 192 pixel display with 15 colors.16k Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation.
External Video input allows subtitling.
NTSC composite video output
SOUND EFFECTS AND MUSIC
Three AY3-8910 Programmable Sound Generators
Nine simultaneous voices
Three independent noise sources
Onboard stereo amplifier drives two 8 ohm speakers
ADDITIONAL I/O CAPABILITIES
Eight analog inputs with 8 bit resolution
Supports four joysticks with pushbutton switches
Eight bit parallel I/O port
Entire unit maps into 256 bytes of memory
DOCUMENTATION AND SOFTWARE
Programming manuals for Video and Sound Processors
Subroutine library and Super Demo Maze Game
Example programs in BASIC, FBASIC and ASSEMBLY
User library and sales support

ARCADE 50 assembled and tested	$325.00
Video and Audio connector set	15.00
4 Joystick connector set	15.00
2 Radio Shack Joysticks	24.00
UHF channel 33 modulator	32.00
Gold Molex connectors	12.00
A/BASIC for 6800	110.00
FBASIC for 6809	110.00
FBASIC (with ARCADE 50)	75.00
FBASIC (manual only)	10.00
ARCADE 80 (TRS Model I)	395.00
ARCADE 100 (S-100 BUSS)	375.00
ARCADE 50 RGB	375.00
LABVIDEO (Motorola EXORbus)	375.00
LABVIDEO RGB	375.00
NEW MV09 6809 Processor Board	225.00
*Comes assembled with PIA and ACIA	
*12 Sockets for 2716, 2732 or RAM	
*Supports DMA disk I/O	
*Ideal for 6809 upgrade or process control	
AMDEK COLOR I Monitor	425.00
AMDEK COLOR II Monitor	799.00
AMDEK COLOR III Monitor	499.00
256K Dynamic Memory Board (assembled)	795.00
256K Dynamic Memory Board (assembled w/64K)	395.00
64K Dynamic Memory Board (assembled)	295.00

Specify 5" or 8" soft sector disk for TSC's FLEX or MICROWARE'S OS/9 system
TERMS: CASH, VISA, MC, C.O.D

FBASIC

TERMINUS DESIGN INC., in conjunction with Microware Systems Corporation, is proud to announce FBASIC-an enhancement of Microware's 6800 A/BASIC. Their fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions, Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50. Arcade statements include:

INIT	MODE	BLANK	BACKDROP
SIZE	MAG	VREG	DELAY
MOVE	DRAW	FCOLOR	JSWITCH
REMOVE	RDRAW	BCOLOR	SWITCH
PSG	TONE	ENVL	VOLUME
ADC	SPRITE	SPNAME	ENDEF
SPCOLOR	RSPRITE	SPDEF	PATDEF
VPEEK	VPOKE	VPRINT	

TERMINUS DESIGN INC
16 SCARBROUGH ROAD
ELLENWOOD, GA 30049
(404) 474-4866

AUTO - COMM

NEW!
For 6809 FLEX

The 'modem' program that automates time sharing communications.

FEATURES:
1) Send text file from personal computer disk to remote mainframe computer.
2) Save incoming text to disk file (verifys acceptance of XON/XOFF controls).
3) Slow transmission mode based on character verify for systems which require speed below baud rate.
4) Eight software selectable UART modes: 8 bit, 7 bit.
5) Self adapts to amount of memory in your computer.
6) Runs in as little as 12K bytes or up to 65K bytes.
7) Reads and writes files of phone numbers to be dialed.
8) Makes any modem a smart modem.

Call Or Write For Further Information
Priced at $75 Manual only $3.95

SYSTEMS Exquire

6712 EAST PRESIDO ST. • SCOTTSDALE, AZ. 85254
VISA-MASTERCARD ACCEPTED
ORDER TOLL FREE 800 272-4817 (IN AZ.) 991-1657
FOR INFORMATION CALL (602) 991-1657

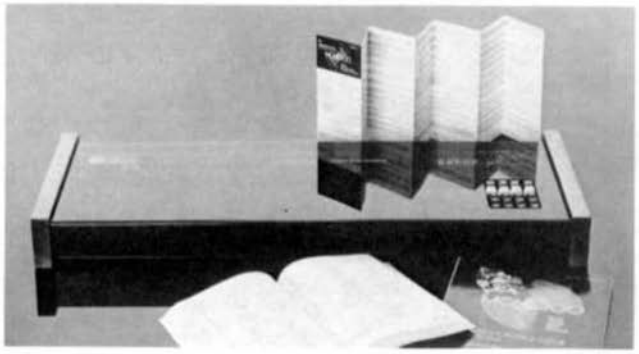
ENGINEERS/TECHNICIANS

THE MICRO 68000 IS DESIGNED FOR YOU!

COMPLETE, READY-TO-GO SYSTEM INCLUDES:

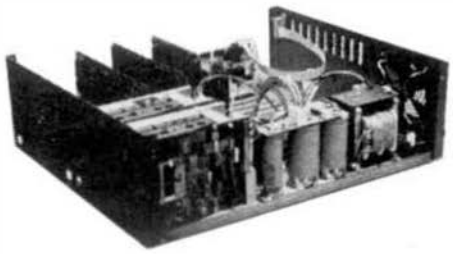
☐ 6 amp switching power supply ☐ Keyboard ☐ Display - Hex & Binary ☐ Pete Bug keyboard monitor ☐ Optional Macs Bug CRT monitor ☐ Attractive cabinet ☐ Dual RS232 interface ☐ 32 bit parallel I/O ☐ Versabus compatibility ☐ The only system that provides for direct entry of 68000 machine code.

CSA *For Information call (619) 566-3911*
Computer System Associates
7562 Trade Street, San Diego, CA 92121



AAA Chicago Computer Center

ELEKTRA
COMPUTER PRODUCTS



ELEKTRA CHASSIS Includes cabinet, 110v power supply, power supply cables, standard disk regulator board with power cables, assembled motherboard with gold square pin connectors $825.00*
ELEKTRA CHASSIS with 2 GIMIX 32K memory boards (64K total) 1300.00*
ELEKTRA CHASSIS, 2 32K GIMIX memory boards, ELEKTRA 8/9 CPU 1550.00*
The $1550.00 combination above with disk controller board and software as indicated (monitor must be added)

	SSB DOS	FLEX™	OS-9™ L1
SSB DCB-4A (SD/DD, controls up to 8 floppies)	2095.00	2245.00***	2295.00***

***Includes SSB DOS also

GIMIX #68 DMA Controller	N/A	2225.00*	N/A
GIMIX #28 Double Density, PIO Controller	N/A	1935.00*	N/A
GIMIX #58 Single Density, PIO Controller	N/A	1865.00*	1975.00*

Other combinations available

ELEKTRA CABINET	260.00
ELEKTRA 110v POWER SUPPLY	175.00*
ELEKTRA CABINET, 110v POWER SUPPLY, AND POWER SUPPLY CABLES	410.00*
ELEKTRA CABINET, 110v PS, PS CABLES, STD. DISK REGULATOR AND CABLES	460.00*

*Add $30.00 for 220v

STANDARD DISK REGULATOR BOARD AND CABLES	50.00
HEAVY DUTY DISK REGULATOR BOARD AND CABLES	75.00
FILLER PLATE FOR 5-1/4" DRIVE OPENING	10.00
FAN FILTER	10.00

See our ad on the next page for our ELEKTRA floppy drive cabinets.

GOLD 10 PIN CONNECTORS (Specify male w/square pins or female)	1.50
TIN 10 PIN CONNECTORS (Specify male w/square pins or female)	.50

	BARE BOARD	KIT	ASSEMBLED
ELEKTRA MOTHERBOARD (Gold connectors)	60.00	320.00	380.00
ELEKTRA MOTHERBOARD (Tin connectors)		240.00	300.00
ELEKTRA CPU 8/9	50.00	225.00	275.00
ELEKTRA CPU 8/9 with baud rate option		250.00	300.00
ELEKTRA DPP DUAL PORT PARALLEL BOARD**	20.00	60.00	80.00
ELEKTRA DPS DUAL PORT SERIAL BOARD**	20.00	60.00	80.00
**CABLE FOR DPP OR DPS (2 needed, specify board)			20.00

MODEMS

	MANUAL ANSWER	AUTO ANSWER
1200 Baud (120 cps), direct connect (U.S. Robotics)	449.00	549.00
300 Baud (30 cps), direct connect (U.S. Robotics)	179.00	219.00
300 Baud (30 cps), acoustic (U.S. Robotics)	149.00	N/A
1200 Baud (120 cps), direct connect (Hayes Smart Modem)		595.00
300 Baud (30 cps), direct connect (Hayes Smart Modem)		229.00

WARNING AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating our kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controllers, drives, and software along with I/O terminals to form working computer systems.

Prices and inventory are subject to change without advance notice.
This ad is our catalog.

SUPER MODEM PROGRAM

Transmit manually to distant computer

Transmit disk file (text) of any length to distant computer

Receive and save disk files (text) of any length on local disk system. If sending computer does not support an X-on/X-off protocol then the received files are limited in size by the computer memory.

Tested to transmit and receive text at speeds up to 9600 baud. (CRT terminal must be capable of operating at a baud rate higher than the one the modem is operated at.)

Half duplex option in case distant computer doesn't echo.

Echo option so user can simulate a time sharing system. (Super Modem Program doesn't support auto-answer but the source is provided for those individuals who want to adapt our program to their special needs.)

Replaces CR with CR/LF (user option) for those using time sharing systems that don't transmit LF's.

Slow disk file transmit (user option) based on character verification for use on time sharing systems to which disk files cannot be sent at speed suggested by the baud rate.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8".
Manual and disk with both source and object code $75.00

STANDARD MODEM PROGRAM

Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Reception of disk files is limited to those small enough to completely fit within the receiving buffer.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8".

Manual with instructions, source listing, and flow chart; disk with both source and object code	$45.00
Manual with instructions, source listing, and flow chart	25.00

TERMS Minimum order $20.00 Shipping and handling estimates within the Continental U.S., add 3% (MINIMUM $2.50). Illinois residents add 6% sales tax. We will refund your overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINIMUM $10.00). Foreign orders must be prepaid in U.S. dollars. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 8 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa and American Express honored.

Our apology. We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The ever frequent changing of our inventory and prices makes it uneconomical to publish a catalog. Our ads are intended to serve that purpose. Prices and inventory are subject to change without advance notice.

AAA CHICAGO COMPUTER CENTER
120 CHESTNUT LANE • WHEELING, IL 60090
(312) 459-0450
Technical consultation available 4 PM to 8 PM most weekdays. Closed evenings and weekends.

ELEKTRA CABINET Made of heavyweight 0.090" thick aluminum. Interior is 18-1/2" wide by 21-7/8" deep by 6-3/4" high. Heavy duty A.C. line cord, A.C. fuse holder, EMI filter. Fan with filter. Back panel has 10 cutouts for 'D' type data connectors. Front panel has key on/off power switch, 2 illuminated push button switches (Reset and NMI/Abort), and two cutouts for 5-1/4" disk drives.

POWER SUPPLY Highest quality linear power supply CONSERVATIVELY rated at 15a @ 5v, 3a @ 16v, 3a @ -16v. 3 primary inputs for light, rated, and heavy loading.

DISK REGULATOR BOARD WITH CABLES Standard version for 2 floppy drives. Heavy duty version for 1 Winchester drive and 1 floppy drive.

ELEKTRA MOTHERBOARD Heavyweight 0.125" thick, 18" long by 9" wide, 11 memory (50 pin) slots. 4 or 8 slots may be cut off for shortening to 14" or 10" lengths respectively. 8 I/O (30 pin) slots. Complete address decoding of I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. Base address for I/O slots can be placed at 32, 64, or 128 byte increments respectively. 1" spacing between all memory and I/O slots. Extended addressing capability for both memory and I/O ports for meeting SS-50C bus specifications. On board baud rate generator with low and high ranges providing jumper selectable rates of 75 through 38,400 for each of the two baud rate lines. Slave device circuitry permitting 1MHz 30 pin disk controllers to run with 2 MHz 50 pin CPU boards.

ELECTRA CPU 8/9 Use either 1 e 6802 or 6808 to run 6800 software or 6809. Has provision for up to 3.7718 E proms, 1K scratchpad, MC6840 triple timer, and an optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. The board supports DMA by either HALT or BUSREQ when a 6809 CPU is used. The board does not support 6 DAT and therefore does not support extended addressing. The board will run any of the MIKBUG™ compatible monitors in the 6802/6808 mode and SBUG-E, HUMBUG, and GMXBUG-09 in the 6809 mode. The ELEKTRA CPU 8/9 will run any of the popular disk controller boards with the appropriate software. OS-9™ Level 1 is available as shown below.

ELEKTRA DPS DUAL PORT SERIAL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 addresses per port with the B port 2 addresses higher than the A port or for 16 addresses per port with the B port 4 addresses higher than the A port. Each port is terminated at two 16 pin dip sockets, one socket configured for modem and the other for terminal or printer. RTS, CTS, DTR, DCD, and DTR are appropriately implemented. Each port has independent selection of baud rate. Each port allows the interrupt request to be independently jumpered to the IRQ or FIRQ/NMI bus line.

ELEKTRA DPP DUAL PORT PARALLEL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be used in either the 4 or 16 addresses per I/O slot configuration occupying the first four addresses of the I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request lines for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line.

SMOOTH™ Software

ALL IN ONE
Editor - Text Processor - Mailing Labels
Mailing Lists - Use any CRT terminal and printer

Supports Editing commands such as bottom, change, delete, find, insert (single line), input (multiple lines), list, next, overlay (with cursor editing, character deletion and insertion), overstrike (for selected darker text), print, restart, set, top, underline, up and verify.

Supports Text Processing commands such as block copy, block move, center, g, margin justification (widen and narrow), paging, and tabbing.

Mailing Lists and Labels. Use the same mailing list disk file (with protected areas) for both mailing labels and repeat letters. Repeat letters are personally adressed to each person or selected persons on the mailing list.

Most Powerful File Handler found in any editor. Append one file to the end of another, or insert (merge) one file into another as designated by the line pointer. Print specified lines to your printer or to a disk file. Edit files larger than the text buffer. Does not produce output files when not desired. Delete disk files from the editor.

Printer commands. Control characters can be sent to the printer for format control either directly from the control terminal or by imbedding them in the text. The set command contains interface initialization and character output routines to support the SWTPC MP-C interface as well as the standard serial and parallel interfaces. Jumps are also provided to user supplied printer routines. User selects the port address (0 thru 7, A or B) thereby eliminating the need for the user to install printer software routines. Editor can be initialized for either 4 or 16 addresses per port.

Editor allows exiting to either the monitor or DOS and then reenter (Warm Start) without destroying previously prepared text in the buffer. The Restart command erases contents in the buffer without the user having to reload the Editor.

The Editor allows the user to toggle between full duplex (no echo) and half duplex (echo) as needed. It responds to commands in both upper and lower case and can be used to create assembler source code and Basic programs as well as text.

Specify 6800 or 6809, SSB or FLEX™, 5" or 8"	75.00
Printed source listing is available for an additional	35.00
All-in-One, Write'n Spell, and Spell'n Fix package	185.00

Software by Technical Systems Consultants, Inc.

Flex™ (includes Editor and Assembler)	150.00
UniFLEX™ (includes one year maintenance and update)	550.00
Editor	50.00
Assembler	50.00
68000 Cross Assembler on 6809	250.00
6809 Cross Assembler on 6800	100.00
Text Processor	75.00
Extended Basic	100.00
Basic Precompiler (specify standard or extended)	50.00
Basic for UniFLEX™ (includes one year maintenance and update)	200.00
Pascal (Flex™)	200.00
Pascal (UniFLEX™) (includes one year maintenance and update)	300.00
Sort/Merge Package	75.00
6809 Flex™ Utilities	75.00
6800 Flex™ Utilities	100.00
Debug Package	75.00
Diagnostic Package	75.00

Software by Microware Systems Corp.	UpDATE	SOURCE	MANUAL O	JECT
OS-9™ Level One Operating System	75.00	400.00	40.00	200.00
OS-9™ Level Two Operating System	75.00	N/A	40.00	500.00
BASIC09™	75.00	N/A	25.00	200.00
BASIC09™ Run-Time Package				100.00
OS-9™ Macro Text Editor		300.00	15.00	125.00
OS-9™ Interactive Assembler		300.00	10.00	125.00
OS-9™ Interactive Debugger (Disk version)		100.00	10.00	50.00
CIS Cobol Compiler	250.00	N/A	80.00	900.00
CIS Cobol Run-Time Package				400.00
Pascal Compiler	100.00	N/A	40.00	400.00
Pascal Run-Time Package				100.00
Microware yearly support service ($200.00 for OS-9 Level 2)				75.00

SWTPC	Kit	Assembled
DMF2C Disk Controller Board		N/A
6809 SWTPC FLEX™ Disk and manual		35.00
6809 SWTPC FLEX™ Disk without manual		15.00
DC-4 Disk Controller	N/A	230.00
SBUG-E (2716 compatible)		25.00
MP-A2 6800 CPU BOARD		150.00
MP-S Serial interface (single port, limited quantity)	40.00	80.00
MP-S2 Serial interface (dual port)	N/A	120.00
MP-LA Parallel interface (dual port, limited quantity)	N/A	N/A
MP-L2 Parallel interface (dual port)	N/A	120.00
MP-R Single voltage 2716 prom programmer	N/A	114.50
MP-N Calculator board	54.95	92.00
MP-T Interrupt timer	N/A	92.00
S32 Universal Static Memory Board	N/A	124.50
MP-09 6809 CPU board (Used $225.00)	N/A	295.00
69A Chassis, P.S., 68B09 CPU, 8K, RAM, One Serial Port	N/A	N/A
Universal 64XX MBT Bare Motherboard. 6800/6809, 4/16 addresses per port, 5.50 pin/8.30 Pin slots, baud rate generator, 6840 interrupt timer slow device circuitry, 15 1/8" x 9 3/8"		55.00
Connectors (10 pin)	Tin	Gold
Male with square pins	0.50	1.50
Female	0.50	1.50

Special Software

4K 6809 HUMBUG	75.00
4K 6800 HUMBUG (RAM needed at $A000 and $D000)	65.00
2K 6800 HUMBUG (With cassette LOAD and PUNCH)	40.00
2K 6800 HUMBUG (Extra commands instead of cassette software)	40.00
Other HUMBUG versions including video versions are available.	
Spell'n Fix by Peter Stark	89.29
Write'n Spell by Peter Stark	75.11
All-in-One, Spell'n Fix, and Write'n Spell package	195.00
Dynamite Disassembler	80.00
SUPER SLEUTH Disassembler System ($101.00 for OS-9 version)	99.00

DISK DRIVES 30 day guarantee, SD/DD capability	1 head	flippy	2 heads
5-1/4", 40 tracks	250.00	260.00	325.00
5-1/4", 80 tracks	325.00	335.00	425.00
MPI — Service Manual (General, covers 40 and 80 track)			20.00
MPI — Service Manual (Specify 40 or 80 track)			25.00
Siemens Manual			10.00
8", 77 tracks	375.00		495.00
8", 77 tracks, Thin-Line	450.00		525.00
Microtime II Calendar and Clock Board			80.00
Data Mart 16K EPROM bareboard (2708 chips)			30.00

SMOOTH™ and **ELEKTRA™** are trademarks of AAA Chicago Computer Center
FLEX and **UniFLEX** are trademarks of Technical Systems Consultants, Inc.
OS-9 and **BASIC09** are trademarks of Microware Systems Corp.
GIMEX® and **GHOST®** are registered trademarks of GIMIX Inc.

ELEKTRA™ SS50 Computer Products

ELEKTRA D-5 Dual drive cabinet for 5-1/4" drives with power supply, line cord, fuse, power switch, and power cable to drives	125.00
ELEKTRA HD-5 (Heavy duty version of D-5 package above)	150.00
ELEKTRA SHD-5 (Super heavy duty. Powers 1 Winchester and 1 floppy)	175.00
5" ribbon cable for dual 5 1/4" disk drives	40.00
ELEKTRA D-8 Dual drive cabinet, power supply, ps cable for 8" drives	350.00
Cabinet for dual 8" drives only	250.00
Power supply for dual 8" drives only	120.00
PS cables only (Specify brand and type of 8" drives)	30.00
6" ribbon cable for dual 8" disk drives	45.00

TERMINALS

Hazeltine 1420	495.00
Hazeltine Espirit 1	525.00
Adds Viewpoint (Green Screen)	525.00
Televideo 925 (Green Screen)	775.00
Televideo 950 (Green Screen)	995.00

Printers

Okidata ML 82A (120 cps, 9x9, oidirectional, serial and parallel)	475.00
Tractor for ML 82A	60.00
Okidata ML 83A (120 cps with tractor)	775.00
2K byte buffer (High speed RS-232)	125.00
Dot addressable graphics	50.00
Okidata ML 84 (200 cps, 2K, Graphics, Parallel)	1195.00
Okidata ML 84 (200 cps, 2K, Graphics, Serial)	1295.00
TI 810 w/lower case and full vertical forms control (limited quantity)	1400.00
Florida Data (600 cps)	3495.00
NEC 3510 Std	1495.00
NEC 7710 Std	2445.00
Epson MX-80FT (Centronics compatible, parallel interface)	525.00
(with Serial RS-232 interface option)	add 75.00
Spare Print Head	39.95
Spare ribbon cartridge	9.50
Epson MX-100	725.00
C. Itoh Prowriter 8510 (Parallel Interface, 120 cps)	525.00

Optimal Technology, Inc. EP-2A-79 Eprom Programmer (Personality Modules extra for above programmer)	169.00
Optimal Technology, Inc. 30 pin parallel I/O board for EP-2A-79	37.00
Software package for EP-2A-79 (Specify 6800 or 6809)	30.00

Smoke Signal Broadcasting

DCB-4A Double Density Controller Board for 5" and 8" with DOS	549.00
SSB DOS Specify 6800 or 6809, BFO or DCB-4A, 5" or 8"	75.00
SSB Monitor (Specify 6800/6809, $8000-$E000-$F7E8)	75.00
SSB version of FLEX™ (without Editor and Assembler)	150.00
LMB-1A Motherboard	399.00
SCB-69 6809 CPU Board	399.00
PAR-1 Dual Port Parallel Card	69.00
SER-2 Dual Port Serial Board with 2 Cables	129.00
Static Memory Boards M-16-X 195.00 M-24-X 295.00 M-32-X 395.00	
Dynamic Memory Boards M-128-X 995.00 M-256-X 1 5.00 M-512-X 1895.00	

GIMIX

6800 CPU Board	224.03
with timers	268.06
with baud rate option	add 30.00
with 2MHz option	add 15.00
2 MHz 6809 Plus CPU, time of day clock, battery backup, 1K NMOS RAM	678.05
CMOS RAM substitution	8.00
GIMIX Dynamic Address Translator	3.00
SwPTC compatible OAT	15.00
9511A Arithmetic Processor (4MHz)	312.00
9512 Arithmetic Processor (3MHz)	265.00
GMXBUG-09 (Terminal Based) 1K scratchpad required	98.65
Manual and Source Listing only	38.82
Bootstrap Prom	30.00
Video Prom (includes bootstrap)	30.00
Filler plate for 5-1/4" drive opening	14.82
Baud rate generator board	88.93
Missing cycle detect card	38.23
Prototyping board (50 pin slot)	56.66
Prototyping board (30 pin slot)	38.33
Disk Controllers (All have data separators and can be used with either single or double headed drives)	
5" single density controller without 1771 chip	156.38
8" single density controller complete	198.48
5" and 8" single density controller complete	226.58
5" double density controller with variable precomp	298.28
OMA 5" AND 8" double density controller with variable precomp	588.68
GIMIX version of FLEX™ (without Editor and Assembler)	90.00
Double disk regulator card	68.22
Ribbon cable for two 5 1/4" disk drives (inboard)	34.96
Ribbon cable for two 8" disk drives (outboard)	44.26
8" disk ribbon cable and back panel connector set	29.25
8" disk drive cabinet with power supply	848.18
19MB Winchester and controller update	4268.90
38MB Winchester and controller update	6888.91

Memory	CMOS WITH BAT. BACKUP	NMOS NO BAT. BACKUP
64K Static RAM Board with 24K of RAM installed	N/A	348.27
64K Static RAM Board with 32K of RAM installed	518.36	398.37
64K Static RAM Board with 48K of RAM installed	N/A	518.47
64K Static RAM Board with 56K of RAM installed	728.58	578.57
64K Static RAM Board with 64K of RAM installed	798.64	638.67
16 Socket EPROM/ROM/RAM Board		238.32
8K Promboard (2708)		98.34
32K Static RAM Board with 32K of RAM installed		245.00

I/O Boards	1 port	2 port	8 port
Serial interface	68.41	128.43	318.46
Parallel interface		88.42	198.45
Cable sets for above boards (specify board)		each	22.95

Video Boards	
64 or 32 X 16	198.71
80 X 24 without RAM character generator	398.74
80 X 24 with RAM character generator	458.76
High resolution (512 X 512 dot resolution)	996.77

2MHz 6809 PLUS Computer System w/o Disk Cont.	w/#58 Cont.	w/#68 Cont.
With 56K of memory	2498.29 2988.59*	3248.49*
With 128K of memory		3798.39
With 128K and 19MB Winchester		8998.09
With 128K, 38MB Winch. dual 8"		17498.99

*Includes GMXBUG/FLEX/OS-9 software selectable

*With CMOS RAM and Battery Backup	add 150.00
Mainframe (Chassis, PS, Switches, Fan, Motherboard, Baud Rate Gen.)	1198.19

AAA Chicago Computer Center

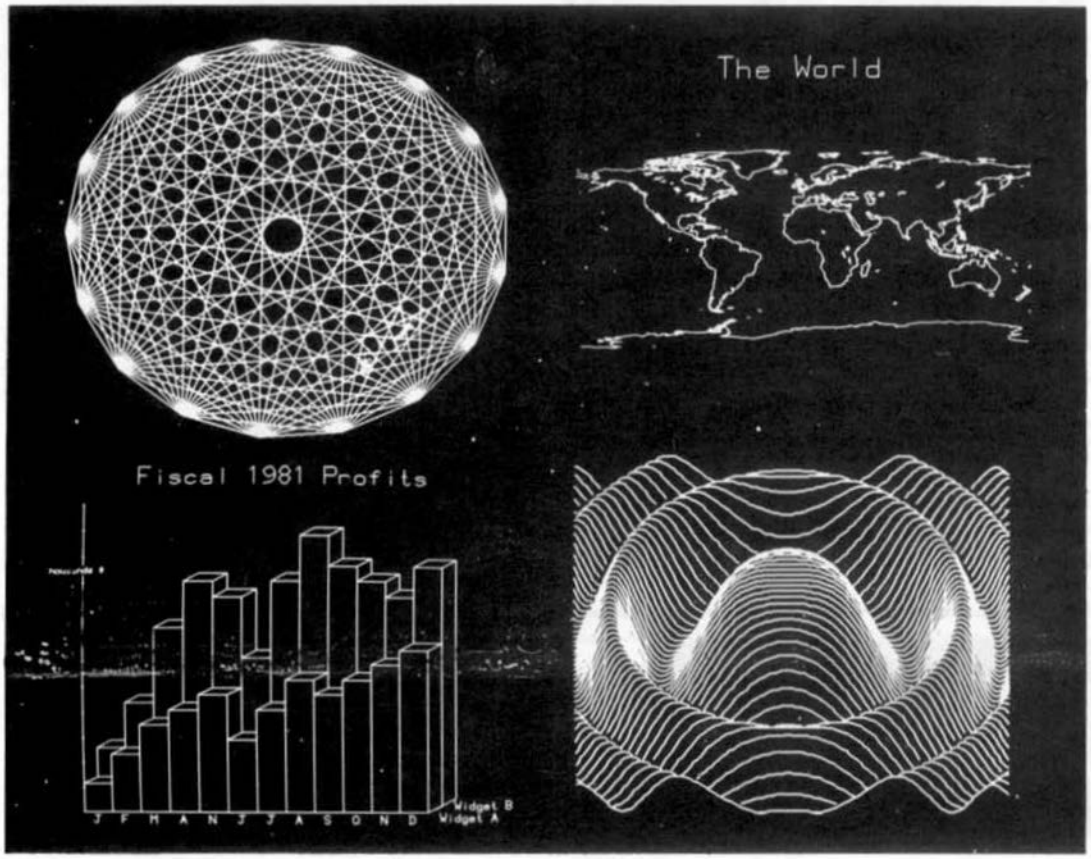
See our ad on the previous page to your left for ordering instructions.

ANNOUNCING ElectroScreen™

only $595

the Superior Alternative

to the Traditional Alphanumeric Terminals



The ElectroScreen™ Intelligent Graphics Board Features:

Graphics

- 512 x 480 resolution bit-mapped display
- Interleaved memory access — fast, snow-free updates

Intelligence

- 6809 on-board mpu
- 6K on-board firmware
- STD syntax high level graphics command set
- Removes host graphics software burden
- Flexible text and graphics integration
- Multiple character sizes
- User programs can be run on-board

Terminal

- Terminal emulation on power-up
- 83 characters by 48 lines display
- Easy switching among user-defined character sets
- Fast hardware scrolling

Additional Features

- SS-50C and SS-64 compatible board
- Board communicates with host through parallel latches
- Composite and TTL level video output
- 8 channel 8 bit A/D converter
- Board occupies 4 address bytes

See your dealer today!

The ElectroScreen manual is available for $10. credited toward purchase of the board.

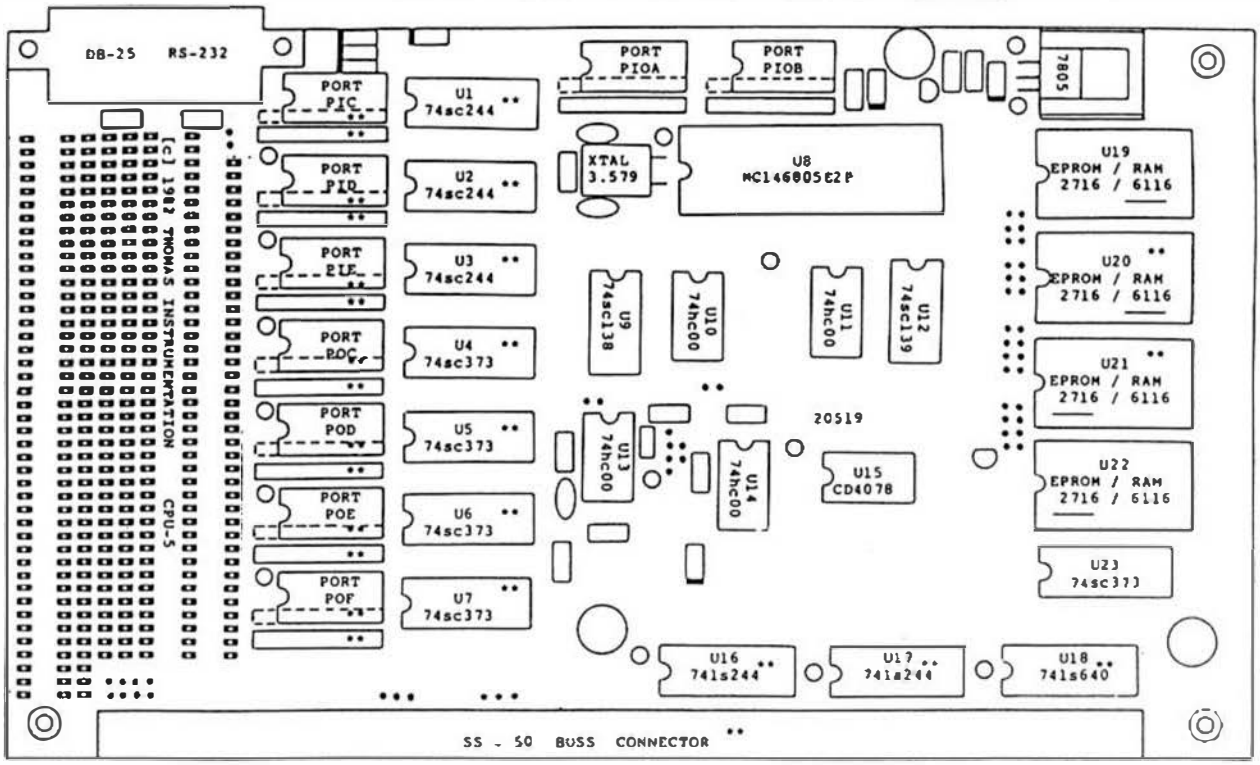
The ElectroScreen has a 90 day warranty from purchase date

Dealers, please contact us for our special introductory package

Privac Inc (703) 671-3900

3711 S. George Mason Dr., Falls Church, Va. 22041

*** NEW 6805 ALL CMOS CARD ***



CPU-5-B	Blank card with documentation	$49.00
CPU-5-P	Partial card (Parts marked ** not incl.)	$229.00] Includes EPROM Monitor
CPU-5-F	Full card with all parts shown (fully assembled)	$349.00] and 115 vac Power Pak.
CROSS-ASSEMBLERS	for FLEX ([c] TSC) 6809 system	$150.00

SOFTWARE

VDISK Treat extended memory as a super fast disk drive

6809 source & object	$149.00
6809 object	$ 99.00

OUTSIDE MODEM PROGRAM incl. source

UniFLEX version	$100.00
FLEX version 6800 & 6809	$ 50.00
HAYES SMART MODEM	$249.00

CROSS ASSEMBLERS for 6800,6801,6805 runs on 6809 FLEX $150.00

TV-EDIT Screen oriented Editor for FLEX 6809 $ 95.00
Please specify 5" or 8" Disk when placing order.

CHIPS

Please note when ordering IC's that a $100.00 Minimum is in effect Orders less than the minimum will be charged an additional $10.00 for handling

6116P-3	2MHZ	$13.50
2016P-2	2MHZ	$10.50
2716	1MHZ	$ 6.50
2716-1	2MHZ	$16.50
146805E2P		$25.00
68B02		$10.00
68B09		$25.00
68B21		$ 6.50
68B40		$12.00
68B50		$ 6.75
74LS640		$ 3.25

HARDWARE

NEW MEMORY PRICES TO INTRODUCE VDISK

S-R/R without memory chips	$120.00
with 8K NMOS@2MHZ	$169.00
with 16K NMOS@2MHZ	$199.00
with 32K NMOS@2MHZ	$299.00
with 48K NMOS@2MHZ	$399.00
Bare Card	$ 49.00

Extender Cards assemb with logic aid

SS-50/50C	$ 35.00
SS-30	$ 25.00
SP-1 Prototype Board A/T	$175.00
SP-1 Bare Card	$ 49.00
SS-50 Wire Wrap Board B/C	$ 39.00
SS-30 Wire Wrap Board B/C	$ 20.00
6802 Super CPU A/T	$235.00
6802 Super CPU B/C	$ 59.00
Video RAM, B/C $49.00, A/T	$195.00
Parallel I/O, B/C $49.00, A/T	$139.00
SS-50 Backplanes 4,6,8,12,&16 position @ $5.00 per slot w/o connectors	
SS-30 8 pos. BP w/o connect.	$ 39.00
Transition Card $49.00, A/T	$ 95.00
Molex Gold Male $1.60, Female	$ 1.60
Molex Tin Male $ .40, Female	$ .50

THOMAS INSTRUMENTATION

168 EIGHTH STREET — AVALON, N.J. 08202 (609) 967-4280
NJ RES. INCLUDE 5% SALES TAX
Allow 3 Weeks for Checks to Clear
CONT. USA INCLUDE $3.00 SHIPPING, CANADA $6.00, FOREIGN $12.00
MASTERCARD, VISA, and C.O.D. ACCEPTED

Send $20.00 to receive full documentation, schematics, & source listings for all boards currently in production.

'68' MICRO JOURNAL ADVERTISERS INDEX

This Index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

THE CHIEFTAIN™ 5¼-INCH WINCHESTER HARD DISK COMPUTER

SO ADVANCED IN SO MANY WAYS . . . AND SO COST-EFFECTIVE . . . IT OBSOLETES MOST OTHER SYSTEMS AVAILABLE TODAY AT ANY PRICE.

● HARD DISK SYSTEM CAPACITY

The Chieftain series includes 5¼- and 8-inch Winchesters that range from 4- to 60-megabyte capacity, and higher as technology advances. All hard disk Chieftains include 64-k memory with two serial ports and DOS69D disk operating system.

● LIGHTNING ACCESS TIME

Average access time for 5¼-inch Winchesters is 70-msec, comparable to far more costly hard disk systems. That means data transfer *ten-times faster* than floppy disk systems.

● 2-MHZ OPERATION

All Chieftains operate at 2-MHz, regardless of disk storage type or operating system used. Compare this to other hard disk systems, no matter *how* much they cost!

● DMA DATA TRANSFER

DMA data transfer to-and-from tape and disk is provided for optimum speed. A special design technique eliminates the necessity of halting the processor to wait for data which normally transfers at a slower speed, determined by the rotational velocity of the disk.

● RUNS UNDER DOS OR OS-9

No matter which Chieftain you select . . . 5¼- or 8-inch floppy, or 5¼- or 8-inch Winchester with tape or floppy back-up . . . they *all* run under DOS or OS-9 with *no need* to modify hardware or software.

● UNBOUNDED FLEXIBILITY

You'll probably never use it, but any Chieftain hard disk system can drive up to 20 other Winchesters, and four tape drives, with a single DMA interface board!

● SMOKE SIGNAL'S HERITAGE OF EXCELLENCE

This new-generation computer is accompanied by the same *Endurance-Certified* quality Dealers and end-users all over the world have come to expect from Smoke Signal. And support, software selection and extremely competitive pricing are very much a part of that enviable reputation.

20-Megabyte Tape Streamer Back-Up Option

Available with all Chieftain hard disk configurations. This cartridge tape capability provides full 20-megabyte disk back-up in less than five minutes with just one command, or copy command for individual file transfers. Transfers data tape-to-disk or disk-to-tape. Floppy back-up is also available in a variety of configurations.

The Chieftain Computer Systems:

Here are the Chieftain 6809-based hard disk computers that are destined to change the data processing industry . . .

☐ **CHIEFTAIN 95W4**
4-megabyte, 5¼-inch Winchester with a 360-k floppy disk drive (pictured).

☐ **CHIEFTAIN 95XW4**
4-megabyte, 5¼-inch Winchester with a 750-k octo-density floppy disk drive.

☐ **CHIEFTAIN 98W15**
15-megabyte, 5¼-inch Winchester with a 1-megabyte 8-inch floppy disk drive.

☐ **CHIEFTAIN 9W15T20**
15-megabyte, 5¼-inch Winchester with a 20-megabyte tape streamer.

Write or call today for details (including the remarkably low prices) on the total Chieftain Series . . . and on dealership opportunities.

SMOKE SIGNAL BROADCASTING®
31336 VIA COLINAS
WESTLAKE VILLAGE, CA 91362
TEL (213) 889-9340

Name _____

Company _____

Address _____

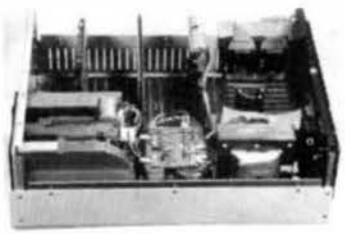
City _____ State _____ Zip _____

Telephone ( ) _____

FLEX - OS-9 LEVEL ONE - UNIFLEX - OS-9 LEVEL TWO

ONLY GIMIX Systems can be configured to run any of these.

GIMIX systems utilize the most powerful 6809 operating systems: FLEX, UniFLEX, OS-9 LEVEL ONE and TWO -- the systems the PROs use. This means a wide selection of software to choose from as well the ability to develop sophisticated, multi-user/multi-tasking programs on your **GIMIX** System.



The GIMIX CLASSY CHASSIS™ consists of a heavy-weight aluminum mainframe cabinet which provides more than ample protection for the electronics and 1 or 2 optional 5¼" drives.

Backpanel connectors can be added for convenient connection of terminals, printers, drives and other peripherals.

A 3 position locking keyswitch enables users to disable the front panel reset button to prevent accidental or unauthorized tampering with the system.

The **GIMIX** system mother board provides fifteen 50 pin slots and eight 30 pin I/O slots -- the most room for expansion of any SS50 system available. The on board baud rate generator features 11 standard baud rates, 75 to 38.4K, for maximum versatility and compatibility with other systems. Extended address decoding allows the I/O block to be addressed anywhere in the 1 megabyte address space. All components feature Gold plated connectors for a lifetime of solid connections. All boards are fully buffered for maximum system expansion.

Each **GIMIX** Mainframe System is equipped with an industrial quality power supply featuring a **ferro-resonant constant voltage transformer** to insure against problems caused by adverse power input conditions such as A.C. line voltage fluctuations etc. The supply provides 8 volts at 30 amps and plus or minus 16 volts at 5 amps, more than enough capacity to power a fully loaded system and two internal drives.

The 2MHz GIMIX 6809 PLUS CPU board includes a time of day clock with battery back-up and 6840 programmable timer to provide the programmer with convenient, accurate time reference. Later addition of 9511 or 9512 arithmetic processors is provided for on the board. The unique **GIMIX** design enables software selection of either OS-9 or FLEX, both included in many complete **GIMIX** systems.

GIMIX STATIC RAM boards require no complicated refresh timing cycles or clocks for data retention. GIMIX memory boards are guaranteed for 2 MHz operation with no wait state or clock stretching required.

Our low power NMOS RAM requires less than 3/4 amp at 8V for a fully populated 64K board. For critical situations, our non-volatile 64K byte CMOS static RAM boards with built in battery back-up retain data even with system power removed. A fully charged battery will power this board for a minimum of 21 days. A write protect switch permits CMOS boards to be used for PROM/ROM emulation and software debugging.

The GIMIX DMA controller leaves the processor free to perform other tasks during disk transfers - an important feature for multi-user/multi-tasking systems where processor time allocation is critical. The DMA board will accomodate up to 4 drives 5¼" or 8" in any combination running single or double density single or double headed. Programmed I/O Disk Controllers are also available.

GIMIX systems are designed with ultimate **RELIABILITY** in mind. You can choose from the below featured systems or select from our wide variety of components to build a custom package to suit your needs.

GIMIX 2MHz 6809 System including: CLASSY CHASSIS, 6809 PLUS CPU BOARD, 56KB STATIC RAM, 2 SERIAL PORTS W/CABLES, GMXBUG MONITOR, FLEX, and OS-9 LEVEL 1 ..... **$3248.49**
FOR TWO 5¼" 40 TRACK DSDD DRIVES ADD ..... **$ 900.00**
GIMIX 128KB WINCHESTER SYSTEM including: CLASSY CHASSIS, 6809 PLUS CPU BOARD, 128KB STATIC RAM, 4 SERIAL PORTS W/CABLES, 5¼" 80 TRACK DSDD FLOPPY DISK DRIVE, 19MB 5¼" WINCHESTER HARD DISK, OS9 LEVEL 2, EDITOR AND ASSEMBLER **$8998.09**

50HZ Versions Available, 8" Drives Available — Contact GIMIX for Prices and Information.

The Sun Never Sets On A GIMIX!

GIMIX users are found on every continent, including Antarctica. A representative group of GIMIX users includes: **Government Research and Scientific Organizations** in Australia, Canada, U.K. and in the U.S.; NASA, Oak Ridge, White Plains, Fermilab, Argonne, Scripps, Sloan Kettering, Los Alamos National Labs, AURA. Universities: Carleton, Waterloo, Royal Military College, in Canada; Trier in Germany; and in the U.S.; Stanford, SUNY, Harvard, UCSD, Mississippi, Georgia Tech. **Industrial users** in Hong Kong, Malaysia, South Africa, Germany, Sweden, and in the U.S.; GTE, Becton Dickinson, American Hoechst, Monsanto, Allied, Honeywell, Perkin Elmer, Johnson Controls, Associated Press, Aydin, Newkirk Electric, Revere Sugar, HI-G/AMS Controls, Chevron. **Computer mainframe and peripheral manufacturers,** IBM, OKI, Computer Peripherals Inc., Qume, Floating Point Systems. **Software houses;** Microware, T.S.C., Lucidata, Norpak, Talbot, Stylo Systems, AAA, HHH, Frank Hogg Labs, Epstein Associates, Softwest, Dynasoft, Research Resources U.K., Microworks, Mela Lab, Computerized Business Systems.

1337 WEST 37th PLACE
CHICAGO, ILLINOIS 60609
(312) 927-5510
TWX 910-221-4055

GIMIX INC.

The Company that delivers

Quality Electronic products since 1975.

FLEX & RS COLOR COMPUTER

If you are tired of playing games on your TRS-80C™ Color Computer, or find that you are handicapped by the limitations of the RS BASIC in trying to write a Program that will allow you to actually USE the Color Computer as a COMPUTER, YOU ARE READY TO MOVE UP TO THE **FLEX9**™ Operating System. If you want to have REAL PROGRAMMING POWER, using an Extremely Powerful Business BASIC, PASCALs, C Compilers, a full-blown Macro Assembler with a Library capability so you are not continuously "reinventing the wheel", YOU ARE READY TO MOVE UP TO THE **FLEX9**™ Operating System. If you would like to see if YOU REALLY COULD USE A COMPUTER IN YOUR BUSINESS, or begin to make your Computer start PAYING IT'S OWN WAY by doing some Computer Work for the millions of small businesses around you, such as Wordprocessing, Payroll, Accounting, Inventory, etc., then YOU ARE READY TO MOVE UP TO THE **FLEX9**™ Operating System. How?? **DATA-COMP** has the way!

DATA-COMP's FLEX9™ Conversion for the TRS-80C™ Color Computer was designed for the SERIOUS COMPUTER USER; with features like greatly increased Display Screens, WITH Lower Case Letters, so you can put a FULL Menu on ONE Screen, or see SEVERAL Paragraphs at the same time; with features like providing a FULL Keyboard so you have FULL Control of your Computer AND it's Programs NATURALLY, without needing a chart to see what Key Combination will give you what function; with USER ORIENTED functions to make using the Operating System natural, like having the Computer AUTOMATICALLY determine what type of Disk is being used in what type of Disk Drive and working accordingly, rather that you have to specify each and every thing for it, or like having the Computer work with the Printer you have been using all along without you having to tell the new Operating System what is there; etc., etc., etc.

DATA-COMP has everything you need to make your TRS-80C™ Color Computer WORK for YOU; from Parts and Pieces to Full, Ready To Use SYSTEMS. DATA-COMP designs, sells, services, and SUPPORTS Computer SYSTEMS, not just Software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

System Requirements

FLEX9 Special General Version x Editor & Assembler (which normally sell for $50.00 ea.)	$150.00
F-MATE(RS) FLEX9 Conversion Rout. for the RS Disk Controller when purchased with Special General FLEX9 Sys.	$69.95
when purchased without the General FLEX9 Sys.	$79.95
Set of Eight 64K RAM Chips w/Mod. Instructions	$99.95
Color Computer with 64K RAM and EXT. BASIC	$549.95
Color Computer with 16K RAM	$375.95
Color Computer with 16K RAM and EXT. BASIC	$465.95

Now Available
Enhanced F-MATE Version 2.1

SPECIAL SYSTEM PACKAGES

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General Version of FLEX9™, F-MATE(RS)™ and a Box of 10 Double Density Diskettes; a COMPLETE, ready to run SYSTEM on your Color TV Set. $1249.95**

DISK DRIVE PACKAGES, etc.

These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable:

PAK #1 — 1 Single Sided, Double Density Sys.	$499.95
PAK #2 — 2 Single Sided, Double Density Sys.	$769.95
PAK #3 — 1 Double Sided, Double Density Sys.	$599.95
PAK #4 — 2 Double Sided, Double Density Sys.	$949.95

PARTS AND PIECES

Radio Shack Disk Controller	$179.95
1 ea. Single Sided, Double Density Disk Drive	$249.95
1 ea. Double Sided, Double Density Disk Drive	$349.95
Single Drive Cabinet with Power Supply	$89.95
Double Drive Cabinet with Power Supply	$109.95
Single Drive Disk Cable for RS Controller	$24.95
Double Drive Disk Cable for RS Controller	$34.95
Micro Tech. Prods., Inc. **LOWER CASE ROM** Adapter	$74.95
Radio Shack BASIC Version 1.1 ROM	$34.95

SOFTWARE

CHESS 6809

Requires FLEX™ and one of the following CRT terminals

Now Runs On Any Type Terminal

Features:

- **Two display boards.**
- **Four levels of play.**
- **Point scoring system.**
- **Play white or black.**
- **Change or set-up boards piece positions.**
- **Forfeit move.**
- **Swap sides.**
- **Make move and swap sides.**
- **Change skill level.**
- **Stop and restart game.**
- **Solve 'Mate in 1-2-3-4' moves.**

$79.95 Specify 5" or 8" disk

This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1600 + .

Note: Personal checks allow 3-4 week delivery.

DIET-TRAC Forecaster

A Diet Planning and Analysis Program

DIET-TRAC Forecaster is a program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C,P,F %) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individuals are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's predicted weight for a 30-day period is printed.

FLEX VERSION	**$59.95**
UniFLEX VERSION	**$89.95**

PRINTERS

The Epson MX-80
$495.00
The Epson MX-100
$725.00

MX-70 $355.00 MX-80 FT $575.00

MEMORY

SWTPC-Motorola, MP32
32K Dynamic Memory Board
Assembled & Tested
1 MHZ - No extended addressing
Can be set up for $0-7FFF or 8000-FFFF

$149.95

MasterCard VISA

DATA-COMP
SOUTH EAST MEDIA

P.O. Box 794 *HIXSON, TN 37343*

1-615-842-4601

Verbatim Diskettes.

5" Soft Sector Disks	
Single Side Single Density	$2.75 ea.
Single Side Double Density	$2.75 ea.
Double Side Double Density	$4.92 ea.
Plastic Storage Box	$2.00 ea.
8" Soft Sector Disks	
Single Side Single Density	$3.75 ea.
Single Side Double Density	$4.10 ea.
Double Side Double Density	$4.75 ea.
Plastic Library Box	$5.00 ea.

Foreign Orders Add 10% Surface—20% Air Mail

WINCHESTER BACKUP UTILITIES

The following utilities allow the backup of any size disk system to any size diskettes.

By simply inserting diskettes as requested by COPYMULT, a large disk system (Winchester, etc.) may be downloaded to your present floppy disk system, any size. No need to fiddle with directory deletions or any of the other tedious operations that must be done using a normal copy routine.

COPYMULT-CMD understands normal "copy" syntax and always keeps up with files already copied by maintaining directories for both host and receiving disk system, thus eliminating hours of tedious keyboard entries and other time consuming cleanup chores.

BACKUP-CMD is a special program that downloads "random" type files, any size.

RESTORE-CMD a special program to restructure copied "random" files for copying, or recopying back to the host system.

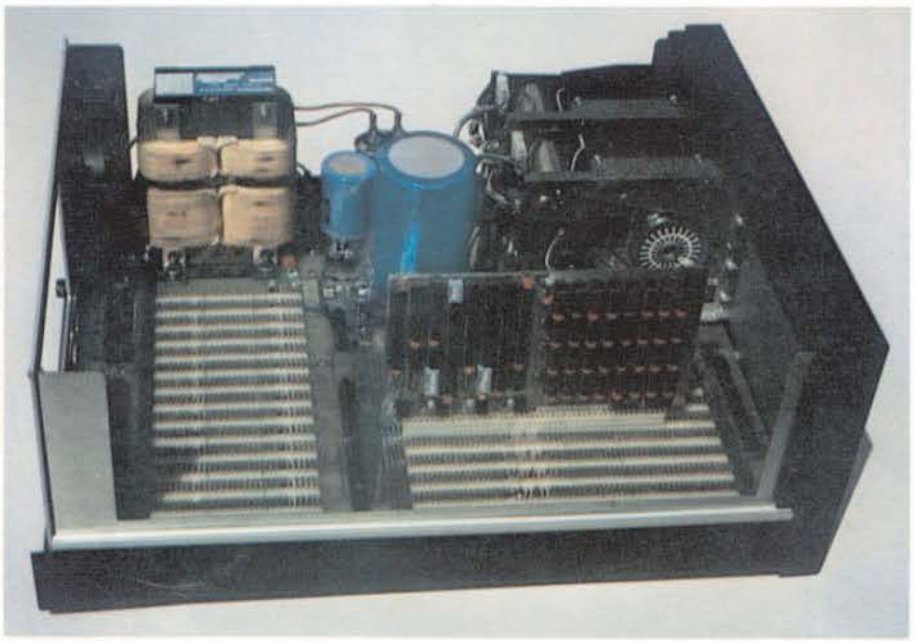
FREELINK-CMD a "bonus" utility that relinks the free chain of a floppy or hard disk thereby eliminating fragmentation.

**** Completely documented source files included.**
**** ALL 3 Programs $99.50 on 8" diskette**

68 Micro Journal
5900 Cassandra Smith Rd.
Hixson, TN 37343

Second Class Postage Paid
At Chattanooga, TN
ISSN 0194-5025

HELIX™



THE MAINFRAME

- Industry Standard Optima™ Cabinet
- Largest Constant Voltage Power Supply in the industry
- S-64 Bus gives 16 Bit Power and S-50 Bus Compatibility
- 10 Main (S-64) Slots
- 14 I/O (S-30) Slots plus 2 On-board
- On-board Baud Rate Generator to 38.4Kb
- Space and Power for two 5¼" Disk Drives
- Full Address Decoding for I/O Slots
- Two RS-232 Serial and Two parallel Ports On-board
- Single Board Construction for Reliability
- Faraday Shielded Bus Lines give "Text Book Clean" Signals

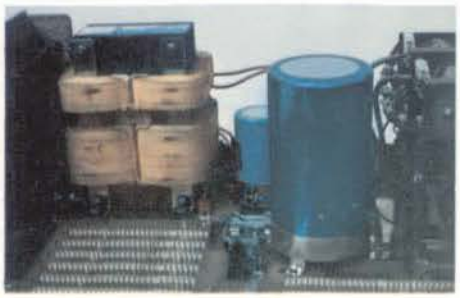
THE PROCESSORS

6809

- Standard 2 MHz Operation
- Standard DAT Compatible with GIMIX and SWTPC
- Standard 6840 Interval Timer
- Standard 1K Scratchpad RAM
- Standard Clock/Calendar with Battery
- Provision for Programmers Console

68000

- Standard 8 MHz Operation
- Memory Management Hardware
- Provision for Programmers Console
- 16 Bit Power and 8 Bit Compatibility



The HELIX™ computer system represents the latest advance in S-50 bus computer systems. Relying on the physical nature of S-50 bus connectors to guarantee compatibility, the HELIX adds 14 bus lines (becoming S-64) to allow a 68000 processor to operate with full 16 bit data transfer and 24 bit addressing, while at the same time providing full interchangability with existing S-50 components.

Offered with a selection of processors, memories, and peripheral controllers, a HELIX system can be configured for applications ranging from advanced hobbyist to multiterminal time-sharing.

Designed to offer the utmost in speed, reliability, and utility at a reasonable price, it represents a new standard of quality for those who require a professionally designed computer for professional use.

THE POWER SUPPLY

- Ferro-resonant Transformer for Line Noise and Under-Voltage Protection
- Conservative 25 Amps at 8.5 Volts
- Conservative 5 Amps at ±16 Volts
- Conservative Component Rating for Reliability

THE COMPONENTS

- Fully Socketed
- Gold Plated Bus Connectors
- Only "B" Series 68XX Components Used
- Only Top Grade Logic Circuits Used
- Industrial Grade Components Throughout

THE MEMORIES

DM-64

- Field Proven
- Proprietary Memory Control Logic
- Fully Transparent Refresh
- Tested at 2.5 MHz Operation

DM-512

- 512K Bytes on a Single S-64 Board
- 16 Bit Power and 8 Bit Compatibility
- Runs in Existing S-50 Systems where Physical Space Allows
- Full 24 Bit Addressing
- Fully Transparent Refresh

THE PRICES

Because of the variety of configurations possible, full pricing cannot be given. Representative prices are:

- 64K 6809 HELIX $1995
- 64K 68000 HELIX $2595
- 512K 6809 HELIX $4450
- 512K 68000 HELIX $4995

HAZELWOOD COMPUTER SYSTEMS

907 E. Terra, O'Fallon, Missouri 63366 (314) 281-1055

Dealer and OEM Inquiries Invited. We support our Dealers.

Optima is a Trademark of Scientific-Atlanta